Lecture Notes in Computer Science 4647

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Ralph Martin Malcolm Sabin
Joab Winkler (Eds.)

Mathematics of Surfaces XII

12th IMA International Conference
Sheffield, UK, September 4-6, 2007
Proceedings

Springer

Volume Editors

Ralph Martin
Cardiff University, School of Computer Science
5 The Parade, Roath, Cardiff CF24 3AA, UK
E-mail: ralph@cs.cf.ac.uk

Malcolm Sabin
Numerical Geometry Ltd.
26 Abbey Lane, Lode, Cambridge CB5 9EP, UK
E-mail: malcolm@geometry.demon.co.uk

Joab Winkler
The University of Sheffield, Department of Computer Science
Regent Court, 211 Portobello Street, Sheffield S1 4DP, UK
E-mail: j.winkler@dcs.shef.ac.uk

Library of Congress Control Number: 2007931446

CR Subject Classification (1998): I.3.5, I.3, I.1, J.6, G.1, F.2, I.4

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-73842-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73842-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12097791 06/3180 5 4 3 2 1 0

Preface

This volume collects the papers accepted for presentation at the 12th IMA Conference on the Mathematics of Surfaces, held at Ranmoor Hall, Sheffield, UK, September, 4–6, 2007. Contributors to this volume include authors from many countries in America, Asia, and Europe. The papers presented here reflect the applicability of various aspects of mathematics to engineering and computer science, especially in domains such as computer-aided design, computer vision, and computer graphics.

The papers in the present volume include eight invited papers as well as a larger number of submitted papers. They cover a range of ideas from underlying theoretical tools to industrial uses of surfaces. Surface types considered range from meshes to parametric and implicit surfaces; some papers investigate general classes of surfaces while others focus more specifically on surfaces such as developable surfaces and Dupin's cyclides. Research is reported on theoretical aspects of surfaces including topology, parameterization, differential geometry, and conformal geometry, and also more practical topics such as geometric tolerances, computing shape from shading, and medial axes for industrial applications. Other specific areas of interest include subdivision schemes, solutions of differential equations on surfaces, knot insertion, surface segmentation, surface deformation, and surface fitting.

We would like to thank all those who attended the conference and helped to make it a success. We are particularly grateful to Lucy Nye at the Institute of Mathematics and its Applications for her hard work in organizing many aspects of the conference, and to Anna Kramer and Frank Holzwarth of Springer for their help in publishing this volume. Following this preface is a list of distinguished researchers who formed the International Programme Committee, and who freely gave their time in helping to assess papers for these proceedings. Due to their work, many of the papers have been considerably improved. Our thanks go to all of them, and to other people upon whom they called to help with refereeing.

June 2007

Ralph Martin
Malcolm Sabin
Joab Winkler

Organization

The Mathematics of Surfaces XII Conference was organized by the Insitute of Mathematics and its Applications (Catherine Richards House, 16 Nelson St., Southend-on-Sea, Essex, England SS1 1EF, UK).

Organizing Committee

Ralph Martin	Cardiff University, UK
Malcolm Sabin	Numerical Geometry Ltd, UK
Joab Winkler	University of Sheffield, UK

Programme Commitee

Chandrajit Bajaj	University of Texas at Austin, USA
Alexander Belyaev	Max Planck Institute for Informatics, Germany
Mathieu Desbrun	Caltech, USA
Gershon Elber	Technion, Israel
Gerald Farin	Arizona State University, USA
Rida Farouki	University of California, Davis, USA
Xiao-Shan Gao	Chinese Academy of Sciences, China
Peter Giblin	University of Liverpool, UK
Ron Goldman	Rice University, USA
Laureano Gonzalez-Vega	University of Cantabria, Spain
Hans Hagen	University of Kaiserslautern, Germany
Edwin Hancock	University of York, UK
Kai Hormann	Technical University of Clausthal, Germany
Shimin Hu	Tsinghua University, China
Bert Jttler	Johannes Kepler University, Austria
Myung-Soo Kim	Seoul National University, Korea
Leif Kobbelt	RWTH Aachen, Germany
Tom Lyche	University of Oslo, Norway
Alberto Paoluzzi	University of Rome 3, Italy
Nick Patrikalakis	MIT, USA
Jorg Peters	University of Florida, USA
Mike Pratt	LMR Systems, UK
Christophe Rabut	INSA Toulouse, France
Paul Sablonniere	INSA Rennes, France
Alla Sheffer	University of British Columbia, Canada
Georg Umlauf	University of Kaiserslautern, Germany
Luiz Velho	IMPA, Brazil
Joe Warren	Rice University, USA
Mike Wilson	University of Leeds, UK

Table of Contents

Regularity Criteria for the Topology of Algebraic Curves and Surfaces 1
Lionel Alberti and Bernard Mourrain

Quadrangle Surface Tiling Through Contouring 29
Pierre Alliez

Surfaces with Piecewise Linear Support Functions over Spherical Triangulations 42
Henrik Almegaard, Anne Bagger, Jens Gravesen, Bert Jüttler, and Zbynek Šír

A Developable Surface of Uniformly Negative Internal Angle Deficit 64
Phillips A. Benton

Rational Maximal Parametrisations of Dupin Cyclides 78
Helmut E. Bez

Discrete Harmonic Functions from Local Coordinates 93
Tom Bobach, Gerald Farin, Dianne Hansford, and Georg Umlauf

Computing the Topology of an Arrangement of Quartics 104
Jorge Caravantes and Laureano Gonzalez-Vega

Non-uniform B-Spline Subdivision Using Refine and Smooth 121
Thomas J. Cashman, Neil A. Dodgson, and Malcolm A. Sabin

Scattered Data Fitting on Surfaces Using Projected Powell-Sabin Splines 138
Oleg Davydov and Larry L. Schumaker

Implicit Boundary Control of Vector Field Based Shape Deformations 154
Wolfram von Funck, Holger Theisel, and Hans-Peter Seidel

Tuning Subdivision Algorithms Using Constrained Energy Optimization 166
Ingo Ginkel and Georg Umlauf

Description of Surfaces in Parallel Coordinates by Linked Planar Regions 177
Chao-Kuei Hung and Alfred Inselberg

Discrete Surface Ricci Flow: Theory and Applications 209
Miao Jin, Junho Kim, and Xianfeng David Gu

Guided C^2 Spline Surfaces with V-Shaped Tessellation 233
Kestutis Karčiauskas and Jorg Peters

MOS Surfaces: Medial Surface Transforms with Rational Domain Boundaries 245
Jiří Kosinka and Bert Jüttler

Mean Value Bézier Surfaces 263
Torsten Langer and Hans-Peter Seidel

Curvature Estimation over Smooth Polygonal Meshes Using the Half Tube Formula 275
Ronen Lev, Emil Saucan, and Gershon Elber

Segmenting Periodic Reliefs on Triangle Meshes 290
Shenglan Liu, Ralph R. Martin, Frank C. Langbein, and Paul L. Rosin

Estimation of End Curvatures from Planar Point Data 307
Xinhui Ma and Robert J. Cripps

Inversion, Degree and Reparametrization for Rational Surfaces 320
Sonia Pérez-Díaz

Discrete Surfaces in Isotropic Geometry 341
Helmut Pottmann and Yang Liu

An Appropriate Geometric Invariant for the C^2-Analysis of Subdivision Surfaces 364
Ulrich Reif

Curvature-Based Surface Regeneration 378
Sebastian T. Robinson and Glen Mullineux

Bounded Curvature Subdivision Without Eigenanalysis 391
Malcolm A. Sabin, Thomas J. Cashman, Ursula H. Augsdorfer, and Neil A. Dodgson

Facial Shape-from-Shading Using Principal Geodesic Analysis and Robust Statistics 412
William A.P. Smith and Edwin R. Hancock

Statistical Methods for Surface Integration 427
William A.P. Smith and Edwin R. Hancock

Skeleton Surface Generation from Volumetric Models of Thin Plate Structures for Industrial Applications 442
Hiromasa Suzuki, Tomoyuki Fujimori, Takashi Michikawa, Yasuhiko Miwata, and Noriyuki Sadaoka

Parallel Tangency in $\mathbb{R}^3$ 465
John P. Warder

Condition Numbers and Least Squares Regression 480
Joab R. Winkler

Propagation of Geometric Tolerance Zones in 3D 494
Song-Hai Zhang, Qi-Hui Zhu, and Ralph R. Martin

Author Index 509

Regularity Criteria for the Topology of Algebraic Curves and Surfaces

Lionel Alberti and Bernard Mourrain

GALAAD, INRIA, BP 93, 06902 Sophia Antipolis, France
mourrain@sophia.inria.fr

Abstract. In this paper, we consider the problem of analysing the shape of an object defined by polynomial equations in a domain. We describe regularity criteria which allow us to determine the topology of the implicit object in a box from information on the boundary of this box. Such criteria are given for planar and space algebraic curves and for algebraic surfaces. These tests are used in subdivision methods in order to produce a polygonal approximation of the algebraic curves or surfaces, even if it contains singular points. We exploit the representation of polynomials in Bernstein basis to check these criteria and to compute the intersection of edges or facets of the box with these curves or surfaces. Our treatment of singularities exploits results from singularity theory such as an explicit Whitney stratification or the local conic structure around singularities. A few examples illustrate the behavior of the algorithms.

1 Introduction

In this paper, we consider the problem of analysing the shape of an object defined by polynomial equations on a bounded domain. Such a problem appears naturally when one has to compute with (algebraic) implicit surfaces [1], but also in algorithms on parameterised curves and surfaces. Typically computing the intersection of two parameterised surfaces leads to the problem of describing or analysing an implicit curve in a 4-dimensional space [2,3].

Our aim is to describe subdivision methods, which given input equations defining such an implicit object, compute a linear approximation of this object, with the same *topology*. The field of application of such methods is Geometric Modeling, where the (semi)-algebraic models used to represent shapes are considered approximations of the *real* geometry. That is, either their coefficients are known with some error or the model itself is an approximation of the actual geometry. In this modelisation process, it is assumed that making the error tend to 0, the representation converges to the actual geometry, at least conceptually. We are going to follow this line, with two specific objectives in mind:

- provide guarantees if possible.
- adapt the computation to the local difficulties of the problem.

Several methods exist to visualize or to mesh a (smooth) implicit surface. Ray tracing techniques [4] which compute the intersection between the ray from the

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 1–28, 2007.
© Springer-Verlag Berlin Heidelberg 2007

eye of the observer and the first object of the scene, produce very nice static views of these surfaces. However isolated singular curves are not well treated and the output of such methods is an image, not a mesh that can be used for other computation.

The famous 'marching cube' method [5,6] developed in order to reconstruct images in 3 dimensions starting from medical data, is based on the construction of grids of values for the function and of sign analysis. It is not adaptive to the geometry of the shape, gives no guarantee of correctness and applies only to smooth surfaces.

Marching polygonizer methods improve the adaptivity of the marching cube by computing only the 'useful' cells [7,8,9], that is those which cut the surface. The algorithm starts from a valid cube (or tetrahedron), and propagate towards the connected cells, which cut the surface. Other variants of the Marching Cube approach have been proposed, to adapt to the geometry of the surface but still with a large number of voxels, even in regions where the surface is very regular. Moreover, the treatment of singularities remains a (open) problem.

Another family of methods called sample methods have also been used. One type uses moving particles on the surface, with repulsion forces which make it possible to spread the particles over the surface [10]. Another type starts from an initial set of sample points on the surface and refine it by inserting new points of the surface, in order to improve the approximation level. Techniques based on Delaunay triangulation of these points have been used for instance for this purpose [11,12].

In the presence of singularities, these methods are not producing correct output and refining the precision parameter of these algorithms increases the number of output points, without solving these singularity problems.

In a completely different direction, methods inspired by Cylindrical Algebraic Decomposition [13] have been proposed to analyse the topology of algebraic curves or surfaces, even in singular cases. The approach has been applied successfully to curves in 2D, 3D, 4D [2,14,15,16,17] and to surfaces [18,19,20]. They use projection techniques based on a conceptual sweeping line/plane perpendicular to some axis, and detect the critical topological events, such as tangents to the sweeping planes and singularities. They involve the exact computation of critical points and genericity condition tests or adjacency tests. The final output of these methods is a topological complex of points, segments, triangles isotopic to the curve or the surface.

They assume exact input equations and rely of the computation of subresultant sequences or calculus with algebraic numbers. This can be a bottleneck in many examples with large degree and large coefficients. Moreover, they are delicate to apply with approximate computation.

In order to combine approximation properties with certification and adaptivity, we consider subdivision methods, which proceed from a large input domain and subdivide it if a regularity criterion is not satisfied. This regularity criterion is designed so that the topology of the curve or the surface lying in the domain can be determined easily. Unfortunately, this type of approach has usually

difficulties when singular points exists in the domain, which make the regularity test failing and the subdivision process going one until some threshold ϵ on the size of the boxes is reached. The obstacle comes from the fact that near a singularity, what ever scale of approximation you choose, the shape or topology of the algebraic objects remains similar. In this paper, we will focus on a regularity criterion which allows to deduce the topology of the variety in a domain, from its intersection with the boundary of the domain. We exploit the local conic structure near points on an algebraic curve or surface, to device algorithms which for a small enough threshold ϵ compute the correct topology, even in the presence of singular points.

These subdivision methods have been already used for solving several equations [21,22]. We recall the recent improvements proposed in [23], which rely on a polynomial solver as the basic ingredient of algorithms for curves and surfaces. Extension of this approach to higher dimensional objects have also been considered [24,14,25,26]. We will recall the subdivision method described in [27] for curves in 3D. It is based on a criterion, which allows us to detect easily when the topology of the curve in a box is uniquely determined from its intersection with the boundary of the box. The treatment of smooth surfaces by subdivision methods as been described in [28]. In this paper, we extend this approach by encompassing the singular case. The approach relies on the computation of the topology of a special curve on the surface, called the polar variety. It is used to detect points at which the surface has a conic structure, meaning we can tell what the topology is by uniquely looking at its intersection with the boundary of a box around the point.

Definitions

Before going into details, here are the notations and definitions we use hereafter:

- For subset domain $S \subset \mathbb{R}^n$, we denote by S° its interior, by $\overline{S}$ its closure, and by ∂S its boundary.
- We call any closed set D such that $\overline{D^\circ} = D$, a domain.
- We call any connected smooth curve $\mathcal{C}$ such that $\mathcal{C} \cap \partial D \neq \emptyset$ and $\mathcal{C} \cap D^\circ \neq \emptyset$, a branch (relative to a domain D), .
- We call any connected submanifold (possibly with boundary) included in the surface (resp. curve), with same the dimension as the surface (resp. curve),a patch of a surface (resp. curve).
- We call a point where the tangent space to the surface (resp. curve) contains the direction x (resp. y, z), a x-critical point (resp. y,z-critical point) of a surface (resp. curve).
- For any point $p \in \mathbb{R}^n$ and $r > 0$, the hypersphere (resp. disk) centered at p of radius r is denoted by $S(p,r) = \{q \in \mathbb{R}^n; \|q - p\| = r\}$ (resp. $D(p,r) = \{q \in \mathbb{R}^n; \|q - p\| \leq r\}$).
- By expressions such as 'topology computation' or 'determine the topology' we mean that we generate an embedded triangulation whose vertices are on the original surface (resp. curve) and which is homeomorphic to that surface

(resp. curve). Our construction actually leads to an embedded triangulation that is isotopic (meaning there is a continuous injective deformation of one onto the other) to the original variety, but this would require some more careful examination of our construction.
- For a box $B = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^3$, its x-faces (resp. y-face, z-face) are its faces normal to the direction x (resp. y, z).

  The size of B, denoted by $|B|$, is $|B| = \max\{|b_i - a_i|; i = 1, \ldots, n\}$.

2 Polynomial Equations

This section recalls the theoretical background of Bernstein polynomial representation and how it is related to the problem we want to solve.

2.1 Bernstein Basis Representation

Given an arbitrary univariate polynomial function $f(x) \in \mathbb{K}$, we can convert it to a representation of degree d in Bernstein basis, which is defined by:

$$f(x) = \sum_i b_i B_i^d(x), \text{ and} \tag{1}$$

$$B_i^d(x) = \binom{d}{i} x^i (1-x)^{d-i} \tag{2}$$

where b_i is usually referred as controlling coefficients. Such conversion is done through a basis conversion [29]. The above formula can be generalized to an arbitrary interval $[a, b]$ by a variable substitution $x' = (b-a)x + a$. We denote by $B_d^i(x; a, b)\binom{d}{i}(x-a)^i(b-x)^{d-i}(b-a)^{-d}$ the corresponding Bernstein basis on $[a, b]$. There are several useful properties regarding Bernstein basis given as follows:

- *Convex-Hull Properties*: As $\sum_i B_d^i(x; a, b) \equiv 1$ and $\forall x \in [a, b], B_d^i(x; a, b) \geq 0$ where $i = 0, ..., d$, the graph of $f(x) = 0$, which is given by $(x, f(x))$, should always lie within the convex-hull defined by the control coefficients $(\frac{i}{d}, b_i)$ [29].
- *Subdivision* (de Casteljau): Given $t_0 \in [0, 1]$, $f(x)$ can be represented by:

$$f(x) = \sum_{i=0}^{d} b_0^{(i)} B_d^i(x; a, c) = \sum_{i=0}^{d} b_i^{(d-i)} B_d^i(x; c, b), \text{ where} \tag{3}$$

$$b_i^{(k)} = (1-t_0) b_i^{(k-1)} + t_0 b_{i+1}^{(k-1)} \text{ and } c = (1-t_0)a + t_0 b. \tag{4}$$

By a direct extension to the multivariate case, any polynomial $f(x_1, \ldots, x_n) \in \mathbb{R}[x_1, \ldots, x_n]$ of degree d_i in the variable x_i, can be decomposed as:

$$f(x_1, \ldots, x_n) = \sum_{i_1=0}^{d_1} \cdots \sum_{i_n=0}^{d_n} b_{i_1,\ldots,i_n} B_{d_1}^{i_1}(x_1; a_1, b_1) \cdots B_{d_n}^{i_n}(x_n; a_n, b_n).$$

where $(B_{d_1}^{i_1}(x_1;a_1,b_1)\cdots B_{d_n}^{i_n}(x_n;a_n,b_n))_{0\le i_1\le d_1,\ldots,0\le i_n\le d_n}$ is the tensor product Bernstein basis on the domain $B := [a_1,b_1]\times\cdots\times[a_n,b_n]\subset\mathbb{R}^n$ and $\mathbf{b} = (b_{i_1,\ldots,i_n})_{0\le i_1\le d_1,\ldots,0\le i_n\le d_n}$ are the control coefficients of f on B. The polynomial f is represented in this basis by the n^{th} order tensor of control coefficients $\mathbf{b} = (b_{i_1,\ldots,i_n})_{0\le i\le d_1,0\le j\le d_2,0\le k\le d_3}$.

De Casteljau algorithm also applies in each of the direction x_i, , $i = 1,\ldots,n$ so that we can split this representation in these directions. We use the following properties to isolate the roots:

This representation provides a simple way to tell the sign of a function in a domain B.

Lemma 2.1. *If all the coefficients $b_{i_1,\ldots,i_n}$ of f in Bernstein basis of $B := [a_1,b_1]\times\cdots\times[a_n,b_n]\subset\mathbb{R}^n$ have the same sign $\epsilon\in\{-1,1\}$, then $\epsilon f(\mathbf{x}) > 0$ for $\mathbf{x}\in B$.*

Proof. As the Bernstein basis elements of the domain B are positive on B and their sum is 1, for $\mathbf{x}\in B$, $f(\mathbf{x})$ is a barycentric combination of the coefficients $b_{i_1,\ldots,i_n}$, of sign ϵ. This $f(\mathbf{x})$ is of sign ϵ.

A consequence is the following interesting property:

Lemma 2.2. *Let f and g by polynomials of degree d_i in x_i $(i = 1,\ldots,n)$ and let $b_{i_1,\ldots,i_n}$ and $c_{i_1,\ldots,i_n}$ be their coefficients in the Bernstein basis of $B := [a_1,b_1]\times\cdots\times[a_n,b_n]$. If $b_{i_1,\ldots,i_n}\le c_{i_1,\ldots,i_n}$ for $0\le i_j\le d_j, j = 1,\ldots,n$ then $f(\mathbf{x})\le g(\mathbf{x})$ for $\mathbf{x}\in B$.*

It will be used in algorithm for computing the topology of implicit curves and surfaces as follows. When the input coefficients of a polynomial f are large rational numbers, instead of working with this expensive arithmetic, we will first compute its coefficients in the Bernstein basis of the given domain B, then normalize them and finally round them up and down to machine precision arithmetic (ie. double). This produces two enveloping functions $\underline{f},\bar{f}$ with the property:

$$\underline{f}(\mathbf{x})\le f(\mathbf{x})\le\bar{f}(\mathbf{x}),\forall\mathbf{x}\in B.$$

These two enveloping polynomials can be used to test sign conditions and regularity criteria, providing certificated results in many situations.

2.2 Univariate Subdivision Solver

Another interesting property of this representation related to Descartes rule of signs is that there is a simple and yet efficient test for the existence of real roots in a given interval. It is based on the number of sign variation $V(\mathbf{b})$ of the sequence $\mathbf{b} = [b_1,\ldots,b_k]$ that we define recursively as follows:

$$V(\mathbf{b}_{k+1}) = V(\mathbf{b}_k) + \begin{cases} 1, \text{if } b_i b_{i+1} < 0 \\ 0, \text{else} \end{cases} \tag{5}$$

With this definition, we have:

Proposition 2.1. *Given a polynomial* $f(x) = \sum_i^n b_i B_i^d(x; a, b)$*, the number* N *of real roots of* f *on* $]a, b[$ *is less than or equal to* $V(\boldsymbol{b})$*, where* $\boldsymbol{b} = (b_i), i = 1, ..., n$ *and* $N \equiv V(\boldsymbol{b}) \mod 2$*.*

With this proposition,

- if $V(\mathbf{b}) = 0$, the number of real roots of f in $[a, b]$ is 0;
- if $V(\mathbf{b}) = 1$, the number of real roots of f in $[a, b]$ is 1.

This yields the following simple and efficient algorithm [30]:

Algorithm 1
INPUT*: A precision* ϵ *and a polynomial* f *represented in the Bernstein basis of an interval* $[b, a]$*:* $f = (\boldsymbol{b}, [a, b])$*.*

- *Compute the number of sign changes* $V(\boldsymbol{b})$*.*
- *If* $V(\boldsymbol{b}) > 1$ *and* $|b - a| > \epsilon$*, subdivide the representation into two sub-representations* $\boldsymbol{b}^-, \boldsymbol{b}^+$*, corresponding to the two halves of the input interval and apply recursively the algorithm to them.*
- *If* $V(\boldsymbol{b}) > 1$ *and* $|b - a| < \epsilon$*, output the* $\epsilon/2$*-root* $(a + b)/2$ *with multiplicity* $V(\boldsymbol{b})$*.*
- *If* $V(\boldsymbol{b}) = 0$*, remove the interval* $[a, b]$*.*
- *If* $V(\boldsymbol{b}) = 1$*, the interval contains one root, that can be isolated with precision* ϵ*.*

OUTPUT: *list of subintervals of* $[a, b]$ *containing exactly one real root of* f *or of* ϵ*-roots with their multiplicities.*

In the presence of a multiple root, the number of sign changes of a representation containing a multiple root is bigger than 2, and the algorithm splits the box until its size is smaller than ϵ.

To analyze the behavior of the algorithm, a partial inverse of Descartes' rule and lower bounds on the distance between roots of a polynomial have been used. It is proved that the complexity of isolating the roots of a polynomial of degree d, with integer coefficients of bit size $\leq \tau$ is bounded by $\mathcal{O}(d^4\tau^2)$ up to some polylog factors. See [31,30] for more details.

Notice that this localization algorithm extends naturally to B-splines, which are piecewise polynomial functions [29].

The approach can also be extended to polynomials with interval coefficients, by counting 1 sign variation for a sign sub-sequence $+, ?, -$ or $-, ?, +$; 2 sign variations for a sign sub-sequence $+, ?, +$ or $-, ?, -$; 1 sign variation for a sign sub-sequence $?, ?$, where $?$ is the sign of an interval containing 0. Again in this case, if a family $\overline{f}$ of polynomials is represented by the sequence of intervals $\bar{\mathbf{b}} = [\bar{b}_0, \ldots, \bar{b}_d]$ in the Bernstein basis of the interval $[a, b]$

- if $V(\bar{\mathbf{b}}) = 1$, all the polynomials of the family $\overline{f}$ have one root in $[a, b]$,
- if $V(\bar{\mathbf{b}}) = 0$, all the polynomials of the family $\overline{f}$ have no roots in $[a, b]$.

The same subdivision algorithm can be applied to polynomials with interval coefficients, using interval arithmetic. This yields either intervals of size smaller than ϵ, which might contain the roots of $f = 0$ in $[a, b]$ or isolating intervals for all the polynomials of the family defined by the interval coefficients.

2.3 Multivariate Bernstein Subdivision Solver

We consider here the problem of computing the solutions of a polynomial system

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ f_s(x_1, \ldots, x_n) = 0 \end{cases}$$

in a box $B := [a_1, b_1] \times \cdots \times [a_n, b_n] \subset \mathbb{R}^n$. The method for approximating the real roots of this system, that we describe now uses the representation of multivariate polynomials in Bernstein basis, analysis of sign variations and univariate solvers (section 2.2). The output is a set of small-enough boxes, which contain these roots. This subdivision solver which can be seen as an improvement of the *Interval Projected Polyhedron* algorithm in [21], it is described in more details in [23].

In the following, we use the Bernstein basis representation of a multivariate polynomial f of the domain $I := [a_1, b_1] \times \cdots \times [a_n, b_n] \subset \mathbb{R}^n$:

$$f(x_1, \ldots, x_n) = \sum_{i_1=0}^{d_1} \cdots \sum_{i_n=0}^{d_n} b_{i_1,\ldots,i_n} \, B_{d_1}^{i_1}(x_1; a_1, b_1) \cdots B_{d_n}^{i_n} x(x_n; a_n, b_n).$$

Definition 2.1. *For any $f \in \mathbb{R}[\mathbf{x}]$ and $j = 1, \ldots, n$, let*

$$m_j(f; x_j) = \sum_{i_j=0}^{d_j} \min_{\{0 \le i_k \le d_k, k \ne j\}} b_{i_1,\ldots,i_n} \, B_{d_j}^{i_j}(x_j; a_j, b_j)$$

$$M_j(f; x_j) = \sum_{i_j=0}^{d_j} \max_{\{0 \le i_k \le d_k, k \ne j\}} b_{i_1,\ldots,i_n} x \, B_{d_j}^{i_j}(x_j; a_j, b_j).$$

Theorem 2.1 (Projection Lemma). *For any $\mathbf{u} = (u_1, \ldots, u_n) \in I$, and any $j = 1, \ldots, n$, we have*

$$m(f; u_j) \le f(\mathbf{u}) \le M(f; u_j).$$

As a direct consequence, we obtain the following corollary:

Corollary 2.1. *For any root $\mathbf{u} = (u_1, \ldots, u_n)$ of the equation $f(\mathbf{x}) = 0$ in the domain I, we have $\underline{\mu}_j \le u_j \le \overline{\mu}_j$ where*

- *$\underline{\mu}_j$ (resp. $\overline{\mu}_j$) is either a root of $m_j(f; x_j) = 0$ or $M_j(f; x_j) = 0$ in $[a_j, b_j]$ or a_j (resp. b_j) if $m_j(f; x_j) = 0$ (resp. $M_j(f; x_j) = 0$) has no root on $[a_j, b_j]$,*
- *$m_j(f; u) \le 0 \le M_j(f; u)$ on $[\underline{\mu}_j, \overline{\mu}_j]$.*

The solver implementation contains the following main steps. It consists in

1. applying a preconditioning step to the equations;
2. reducing the domain;
3. if the reduction ratio is too small, to split the domain

until the size of the domain is smaller than a given epsilon.

The following important ingredients of the algorithm parameterize its implementation:

Preconditioner. It is a transformation of the initial system into a system, which has a better numerical behavior. Solving the system $\mathbf{f} = 0$ is equivalent to solving the system $M\,\mathbf{f} = 0$, where M is an $s \times s$ invertible matrix As such a transformation may increase the degree of some equations, with respect to some variables, it has a cost, which might not be negligible in some cases. Moreover, if for each polynomial of the system not all the variables are involved, that is if the system is sparse with respect to the variables, such a preconditioner may transform it into a system which is not sparse anymore. In this case, we would prefer a partial preconditioner on a subsets of the equations sharing a subset of variables. We consider Global transformations, which minimize the distance between the equations, considered as vectors in an affine space of polynomials of a given degree and Local straightening (for $s = n$), which transform locally the system $\mathbf{f}$ into a system $J^{-1}\mathbf{f}$, where $J = (\partial_{x_i} f_j(\mathbf{u}_0))_{1\le i,j\le s}$ is the Jacobian matrix of $\mathbf{f}$ at a point $\mathbf{u}$ of the domain I, where it is invertible.

It can be proved that the reduction based on the polynomial bounds m and M behaves like Newton iteration near a simple root, that is we have a quadratic convergence, with this transformation.

Reduction strategy, that is the technique used to reduce the initial domain, for searching the roots of the system. It can be based on Convex hull properties as in [21] or on Root localisation, which is a direct improvement of the convex hull reduction and consists in computing the first (resp. last) root of the polynomial $m_j(f_k; u_j)$, (resp. $M_j(f_k; u_j)$), in the interval $[a_j, b_j]$. The current implementation of this reduction steps allows us to consider the convex hull reduction, as one iteration step of this reduction process.

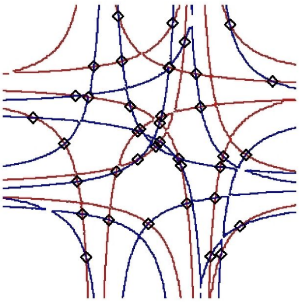
The guarantee that the computed intervals contain the roots of f, is obtained by controlling the rounding mode of the operations during the de Casteljau computation.

Subdivision strategy, that is technique used to subdivide the domain, in order to simplify the forthcoming steps, for searching the roots of the system. The approach, that we are using in our implementation is the parameter domain bisection: The domain b is then split in half in a direction j for which $|b_j - a_j|$ is maximal. But instead of choosing the size of the interval as a criterion for the direction in which we split, we may choose other criterion depending also on the value the functions m_i, M_j or f_j (for instance where $M_j - m_j$ is maximal).

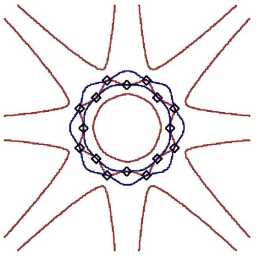
A bound for the complexity of this method is detailed in [23]. It involves metric quantities related to the system $\mathbf{f} = 0$, such as the Lipschitz constant of $\mathbf{f}$ in B, the entropy of its near-zero level sets, a bound d on the degree of the equations in each variable and the dimension n.

2.4 Example

Here are some comparisons of the different strategies, describing the number of iterations in the main loop, the number of subdivision of a domain, the number of boxes produced by the method, the time it takes. We compare the method **sbd**,



method	iterations	subdivisions	results	time (ms)
sbd	4826	4826	220	217
rd	2071	1437	128	114
sbds	3286	3286	152	180
rds	1113	748	88	117
rdl	389	116	78	44



method	iterations	subdivisions	results	time (ms)
sbd	84887	84887	28896	3820
rd	82873	51100	20336	4553
sbds	6076	6076	364	333
rds	1486	920	144	163
rdl	1055	305	60	120

Fig. 1. Behavior of the subdivision solver for different preconditioners

a pure subdivision approach, **rd** a method doing first reduction and based on a univariate root-solver using the Descarte's rule. **sbds**], a subdivision approach using the global preconditioner, **rds**, a reduction approach using the global preconditioner, **rdl**, a reduction approach using the jacobian preconditioner (see Fig. 1). The first example is a bivariate system, with equations of degree 12 in each variable, the second example is of bidegree $(8, 8)$. For more details on this solver, see [32,23].

3 Planar Curves

In this section, we consider a curve $\mathcal{C}$ in $\mathbb{R}^2$, defined by the equation $f(x, y) = 0$ with $f \in \mathbb{Q}[x, y]$ and a domain $B = [a, b] \times [c, d] \subset \mathbb{R}^2$.

3.1 Regularity Criterion for Planar Curves

We recall that a tangent to the curve $\mathcal{C}$ is a line, which intersects $\mathcal{C}$ with multiplicity ≥ 2. In particular, any line through a singular point of $\mathcal{C}$ is tangent to $\mathcal{C}$.

Definition 3.1. *We say that the curve $\mathcal{C}$ is y-regular (resp. x-regular) in B, if $\mathcal{C}$ has no tangent parallel to the y-direction (resp. x-direction) in B.*

Notice that if $\mathcal{C}$ is x-regular (or y-regular) it is smooth in B since it cannot have singular points in B. A curve is *regular* in B, if it is x-regular *or* y-regular in B.

We are going to show that if $\mathcal{C}$ is x-regular in B, then its topology can be deduced from its intersection with the boundary ∂B.

Definition 3.2. *For a point $p \in \mathcal{C} \cap \partial B$, we define its interior tangent $T_p^i(\mathcal{C})$ as the tangent of $\mathcal{C}$ at p, pointing inside B.*

If the curve is not tangent to ∂B at $p \in \partial B$ and p is not a corner point of B, this direction is defined by $\epsilon_p T_p(\mathcal{C})$ with $\epsilon_p = \text{sign}(T_p(\mathcal{C}) \cdot \nu_p)$, where ν_p is the unit normal interior vector to ∂B at p.

If the curve is tangent to ∂B at p, we say that the point is of multiplicity 2. In this case, we consider the half branches of $\mathcal{C}$ at p, and associate to each of them the unit tangent vector to this branch, if the branch is inside B near p. In the following, if two opposite unit vectors are attached to a point p, we will duplicate this point, so that a point is attached to a unique interior tangent.

If p is at the corner of B, we extend this definition, as follows: We consider the cone of interior normal vectors ν_p of B at p, and require that $T_p^i(\mathcal{C}) \cdot \nu_p \geq 0$ for all the vectors ν_p in this cone. Thus, this interior tangent might not exist for corner points.

Definition 3.3. *For a point $p \in \mathcal{C} \cap \partial B$ with interior tangent $T_p^i(\mathcal{C})$, we define its x-index (resp. y-index) as $\text{sign}(T_p^i(\mathcal{C}) \cdot e_1)$ where e_1 is the unit vector $(1,0) \in \mathbb{R}^2$ (resp. $(0,1) \in \mathbb{R}^2$. If the interior tangent of $\mathcal{C}$ at p does not exist, we define the x-index of p as 0.*

For a y-regular curve $\mathcal{C}$ (with no vertical tangent) in B and $p \in](a,c),(a,d)[$, we have $x\text{-index}(p) = 1$. If $p \in](b,c),(b,d)[$, its x-index is -1. Moreover, if the curve is not tangent to the horizontal segment on the boundary of B, the x-index of a point of $\mathcal{C} \cap \partial P$ which is not a corner point of B is not 0.

Lemma 3.1. *If $\mathcal{C}$ is y-regular in B, then a branch of $\mathcal{C} \cap B$ connects a point p of x-index 1 to a point q of x-index -1, such that $x_p < x_q$.*

Proof. As the curve is y-regular, it has no vertical tangent and thus no closed loop in B. Consequently, each of the interior connected components of $\mathcal{C} \cap B$ intersects ∂B in two distinct points $p, q \in \mathcal{C} \cap \partial B$ (with $x_p \leq x_q$).

Assume that the x-index of p, q are the same. Suppose that this index is 1. Then for an analytic parameterisation $s \in [0,1] \mapsto (x(s), y(s))$ of the branch $[p,q]$ with $(x(0), y(0)) = p$, $(x(1), y(1)) = q$, we have $\partial_s x(0) > 0$, $\partial_s x(1) < 0$. This implies that for a value $0 < s_0 < 1$, $x(s_0) > x(1) = x_q \geq x(0) = x_p$ and that there exists $s_0' \in]0,1[$ such that $x(s_0') = x(1)$. We deduce that $\partial_s x(s)$ vanishes in $[0,1]$ and that the branch $[p,q]$ of $\mathcal{C}$ has a vertical tangent, which is excluded by hypothesis. If the index of p and q is -1, we exchange the role of p and q and obtain the same contradiction. As $\partial_s x(s) > 0$ for $s \in [0,1]$, we have $x_p < x_q$, which proves the lemma.

Lemma 3.2. *Suppose that $\mathcal{C}$ is y-regular in B and let p, q be two consecutive points of $\mathcal{C} \cap \partial B$ with*

- *x-index$(p) = 1$, x-index$(q) = -1$,*
- *$x_p < x_q$.*

Then p, q belong to the same branch of $\mathcal{C} \cap B$.

Proof. Let p, q be two consecutive points of $\mathcal{C} \cap \partial B$, with x-index$(p) = 1$, x-index$(q) = -1$ and $x_p < x_q$. Suppose that p, q belong to two branches (p, p'), (q, q') of $\mathcal{C}$ in B. As the curve $\mathcal{C}$ is smooth in B, the two branches do not intersect, and the points q, q' are on the same component of $\partial B - \{p, p'\}$. As x-index$(q) = -1$, by the previous lemma x-index$(q'') = 1$ and $x(q') < x(q)$. This implies that q' is a point in-between p and q on ∂B, on one component of $\partial B - \{p, q\}$. A similar argument shows that p' is a point between p and q on ∂B on the other component of $\partial B - \{p, q\}$. This contradicts the fact that p and q are consecutive points of ∂B.

Proposition 3.1. *If $\mathcal{C}$ is regular in B, its topology in B is uniquely determined by its intersection $\mathcal{C} \cap \partial B$ with the boundary of B.*

Proof. Exchanging the role of x and y if necessary, we can assume that $\mathcal{C}$ is y-regular. We prove the proposition by induction on the number $N(\mathcal{C})$ of points on $\mathcal{C} \cap \partial_B$, with non-zero x-index. We denote this set of points by $\mathcal{L}$.

Since the curve has no vertical tangent in B and has no closed loop, each of the connected components of $\mathcal{C} \cap B$ intersects ∂B in two distinct points of x-index $\neq 0$. Thus if $N(\mathcal{C}) = 0$, then there is no branch of $\mathcal{C}$ in B.

Assume now that $N(\mathcal{C}) > 0$, and let us show that it is possible to find two consecutive points p, q of $\mathcal{L}$ with x-index$(p) = 1$, x-index$(q) = -1$, $x_p < x_q$. As the curve $\mathcal{C}$ is smooth in B, its k branches are not intersecting each other and they split B into $k+1$ connected components, which intersect the boundary ∂B. Consider a branch $[p, q]$ which separates k of these components from the last one. Then, there is no other points of $\mathcal{C} \cap \partial B$ in-between p and q. By lemma 3.2, we have x-index$(p) = 1$, x-index$(q) = -1$, $x_p < x_q$.

Removing this branch from $\mathcal{C}$, we obtain a new curve $\mathcal{C}'$ which is still y-regular and such that $N(\mathcal{C}') < N(\mathcal{C})$. We conclude by induction hypothesis, that the topology of $\mathcal{C}'$ and thus of $\mathcal{C}$ is uniquely determined.

If $\mathcal{C}$ is y-regular, by exchanging the role of x and y, we can deduce the topology in B from the points on ∂B. The initial domain can be subdivided in such a way, that each box contains either a unique singular point and is small enough or is regular. For the small boxes B containing a unique singular point, we are going to connect the center of B with the points of $\mathcal{C} \cap \partial B$. For the regular boxes, we apply the following algorithm:

Algorithm 2 (Connection for a regular curve)
INPUT: *an algebraic curve $\mathcal{C}$ and a domain $B = [a, b] \times [c, d] \subset \mathbb{R}^2$ such that $\mathcal{C}$ has no vertical (resp. horizontal) tangent in B.*

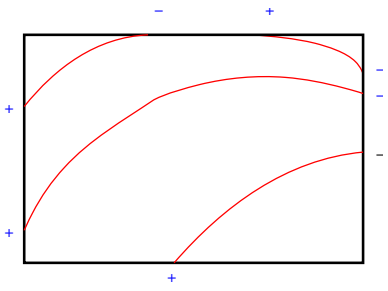


Fig. 2. Illustration of Proposition 3.1

- *Isolate the points $\mathcal{C} \cap \partial B$ and compute their x-index (or y-index);*
- *Order the points of $\mathcal{C} \cap \partial B$ with non-zero x-indices (resp. y-indices); clockwise and store them in the circular list $\mathcal{L}$.*
- *While $\mathcal{L}$ is not empty,*
  - *find two consecutive points p, q in $\mathcal{L}$ with x-index$(p) = 1$, x-index$(q) = -1$, $x_p < x_q$ (resp. y-index$(p) = 1$, y-index$(q) = -1$, $y_p < y_q$);*
  - *add the arc $[p, q]$ to the set $\mathcal{B}$ of branches and remove p, q from $\mathcal{L}$.*

OUTPUT: *the set $\mathcal{B}$ of branches of $\mathcal{C}$ in B.*

3.2 Tests of Regularity

Here are simple tests to check the regularity of a curve in a domain, which extends in some way the criterion in [33].

Proposition 3.2. *If $\partial_y f(x, y) \neq 0$ (resp. $\partial_x f(x, y) \neq 0$) in a domain $B = [a_0, b_0] \times [a_1, b_1] \subset \mathbb{R}^2$, the curve $\mathcal{C}$ is regular on B.*

Proof. If $\partial_y f \neq 0$ in B, the curve $\mathcal{C}$ cannot have vertical tangent and is thus y-regular.

The representation of polynomials in Bernstein basis can be used to verify easily this condition.

Proposition 3.3. *If the coefficients of $\partial_y f(x, y) \neq 0$ (resp. $\partial_x f(x, y) \neq 0$) in the Bernstein basis of the domain $B = [a, b] \times [c, d] \subset \mathbb{R}^2$ have the same sign $\in \{-1, 1\}$, then the curve $\mathcal{C}$ is regular on B.*

Proof. If the Bernstein coefficients of $\partial_y f$ (resp. $\partial_x f$) are all > 0 (resp. < 0), so is $\partial_y f$ (resp. $\partial_x f$) in B (by the convex hull property), which implies that the curve $f(x, y) = 0$ is regular in B.

Suppose that $\partial_y f(x, y) \neq 0$ in B. Then two branches of $\mathcal{C}$ in B cannot have points with the same x-coordinate (otherwise $\partial_y f$ would vanish in-between these two points). Consequently, the branches do not overlap by projection on the x-axis and the connection algorithm can be simplified as follows:

– Compute the points of $\mathcal{C} \cap \partial B$, repeating a point if its multiplicity is even.
– Sort them by lexicographic order so that $x > y$: $\mathcal{L} := \{p_1, p_2, \ldots\}$
– Connect them by pair $[p_1, p_2], [p_3, p_4], \ldots$ of consecutive points in $\mathcal{L}$.

3.3 Algorithm of Subdivision

This yields the following new subdivision algorithm for a planar implicit curve:

Algorithm 3 (Topology of a planar implicit curve)
INPUT*: A domain $B_0 \subset \mathbb{R}^2$, a curve $\mathcal{C}$ defined by the squarefree polynomial equation $f(x, y) = 0$ with $f \in \mathbb{Q}[x, y]$ and $\epsilon > 0$.*

– *Compute the x and y critical points of $f(x, y) = 0$.*
– *$\mathcal{L} = \{B_0\}$*
– *While $\mathcal{L}$ is not empty,*
  - *Choose and remove a domain B of $\mathcal{L}$;*
  - *Test if there is a unique critical point, which is not singular in B;*
  - *If it is the case, compute the topology of $\mathcal{C}$ in B (Algorithm 2),*
  - *else if $|B| > \epsilon$, subdivide the box into subdomains and add them to $\mathcal{L}$,*
  - *otherwise connect the center of the box to the points of $\mathcal{C} \cap \partial B$.*

OUTPUT*: a set of points and a set of (smooth) arcs connecting these points.*

An interesting advantage of this algorithm compared to sweeping algorithms (such as [15]) is that it avoids a projection on a line, which requires (sub)-resultant computations and the manipulation of algebraic numbers.

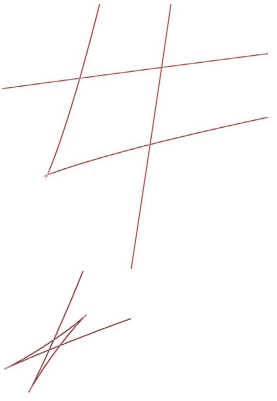
A variant of this algorithm consists in filtering the regularity test by sign tests (Proposition 3.3) on the Bernstein representation of f in B and to compute the x and y critical points only in domains B where the Bernstein coefficients of f and $\partial_x f, \partial_y f$ have sign changes.

Proposition 3.4. *For $\epsilon > 0$ small enough, the algorithm 3 computes a graph of points which is isotopic to the curve $\mathcal{C} \cap B$.*

Proof. By Proposition 3.2, if the box B contains no singular points and at most one (smooth) x or y critical point, the topology of the curve is uniquely determined by its intersection with ∂B. In these domains, the algorithm produce the correct topological graph.

Suppose that the algorithm treats a domain B which contains a singular point p. Then its size is smaller than ϵ. We are going to show that for ϵ small enough, the topology of the curve in B is given by the graph connecting a point inside to the points of $\mathcal{C} \cap \partial B$. Let us first choose ϵ' smaller than half the minimal distance between two distinct x or y critical points of $\mathcal{C}$ so that B contains only one singular point p of $\mathcal{C}$.

Using Puiseux expansion at the singular point p [34,35], we can construct a disk $D(p, \epsilon')$ of radius ϵ', centered at p, in which the curve $\mathcal{C}$ is the union of real branches described by the image of analytic functions σ_i, $i = 1, \ldots, r$ over the



This curve is the discriminant curve of a bivariate system with few monomials used in [37] to give a counter-example to Kushnirenko's conjecture. It is of degree 47 in x and y, and the maximal bit size of its coefficient is of order 300. It takes less that 10s to compute the topological graph, by rounding up and down to the nearest double machine precision numbers, applying the subdivision techniques on the enveloping polynomials. We observe a very tiny domain, which at first sight looks like a cusp point, but which contains in reality, 3 cusps points and 3 crossing points. The central region near these cusp points is the region where counter-examples have been found.

Fig. 3. A discriminant curve

interval $[0,1]$, with $\sigma_i(0) = p$ and $\sigma_i(1) \in \partial D(p, \epsilon')$. Since in this disk, the only critical point of $\mathcal{C}$ is p, these branches have no tangents parallel to the x or y-axis on the interval $]0,1[$.

Let us choose a box B of size $< \epsilon = \frac{1}{\sqrt{2}}\epsilon'$ with $p \in B^\circ$. Then $B \subset D(p, \epsilon')$. As $\sigma_i(0) = p \in B^\circ$ and $\sigma_i(1) \in \partial D(p, \epsilon')$, each branch intersects ∂B. Suppose now that a given branch intersects ∂B in 2 or more points $\sigma_i(t_0)$ $\sigma_i(t_1)$ with $0 \leq t_0 < t_1 \leq 1$. Then there is a value $t_0 \leq t \leq t_1$ such that the tangent at t is parallel to one of the x or y directions, which is excluded. This implies that the number of real branches at p and the number of points of $\mathcal{C} \cap \partial B$ coincides for a B of size $< \epsilon$, with $p \in B^\circ$.

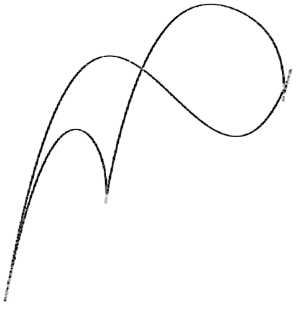
Consequently, for ϵ small enough, the algorithm compute the correct topological graph.

Remark 3.1. The number of branches at a singular point can be computed from information on the boundary of the domain by using the topological degree [36]. Thus, it is possible to control effectively when the number of points of $\mathcal{C} \cap \partial B$ is the number of branches at the singular point p and in this way, to certify the result of this algorithm. But describing these techniques would lead us too far outside the scope of this paper.

Examples are shown in Figures 3 and 4.

4 Space Curves

In this section, we consider a curve $\mathcal{C}$ of $\mathbb{R}^3$. Suppose that $I(\mathcal{C}) = (f_1, f_2, \ldots, f_k)$ and for two polynomials $f(x,y,z), g(x,y,z) \in I(\mathcal{C})$, we define $\mathbf{t} = \bigtriangledown(f) \wedge \bigtriangledown(g)$. We are interested in the topology of $\mathcal{C}$ in a box $B = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2]$. Similar to the 2D case, we can represent f, g and each component of $\mathbf{t}$ in the Bernstein basis for the domain B. As we will see, the sign changes of the resulting Bernstein coefficients will make it possible to test the regularity of the curve with minimal effort.



This curve is the projection onto the (x, y) plane of the curve of points with tangent parallel to the z-direction for a surface of degree 4. It is defined by the equation of degree 12, and has 4 real cusps and 2 real crossing points. The size of the coefficients is small and the topological graph is computed in less than 1 second. It defines 4 connected regions in the plane, one of these being very small and difficult to see on the picture.

Fig. 4. The apparent contour of a quartic surface

4.1 Regularity Criterion for Curves in 3D

In this section, we recall some criteria for the regularity of space curves. More details can be found in [27]. These criteria are based on the following property:

Proposition 4.1. *If $\mathcal{C}$ is smooth in B and if for all $x_0 \in \mathbb{R}$, the plane $x = x_0$ has at most one intersection point with the curve $\mathcal{C}$ in B, then the topology of $\mathcal{C}$ is uniquely determined from the points $\mathcal{C} \cap \partial B$.*

Proof. Consider the projection $\pi_z(\mathcal{C})$ of the curve $\mathcal{C}$ in B along the z direction. Then the components of $\mathcal{C}$ in B projects bijectively on the (y, z) plane. Otherwise, there exist two points p_0 and p_1 lying on $\mathcal{C}$ such that $\pi_z(p_0) = \pi_z(p_1) = (x_0, y_0)$, then p_0 and p_1 belong to $x = x_0$ which are functions of the form $y = \Phi(x)$. Otherwise, there exist two points on $\pi_z(\mathcal{C})$ and (and on $\mathcal{C} \cap B$) with the same x-coordinate. Consequently, for $x \in [a_0, b_0]$ there is at most one branch of $\pi_z(\mathcal{C})$ in B above x, and the connected components of $\mathcal{C} \cap B^\circ$ project bijectively onto non-overlapping open intervals of $[a_0, b_0]$ as $\pi_z(\mathcal{C})$ does. We conclude as in the 2D case (proposition 3.3), by sorting the points of $\mathcal{C} \cap \partial B$ according to their x-coordinates, and by grouping them in consecutive pairs corresponding to the start and end points of the branches of $\mathcal{C} \cap B$.

Here is an explicit way to check the conditions of Proposition 4.1:

Proposition 4.2. *The 3D spatial curve $\mathcal{C}$ defined by $f = 0$ and $g = 0$ is regular on B, if*

- $\boldsymbol{t}_x(\boldsymbol{x}) \neq 0$ *on B, and*
- $\partial_y h \neq 0$ *on z-faces, and $\partial_z h \neq 0$ and it has the same sign on both y-faces of B, for $h = f$ or $h = g$.*

Proof. Let us fix $x_0 \in [a_0, b_0]$ where $B = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2]$, let $U = \{x_0\} \times [a_1, b_1] \times [a_2, b_2]$ and let $\Phi_{x_0} : (x_0, y, z) \in U \mapsto (f(x_0, y, z), g(x_0, y, z))$. We are going to prove that under our hypothesis, Φ_{x_0} is injective. The Jacobian $\mathbf{t}_x(x_0, y, z)$ of Φ_{x_0} does not vanish on U, so that Φ_{x_0} is locally injective. We consider the level-set $f(\mathbf{x}) = f_0$ for some $f_0 \in f(U)$. It cannot contain a closed loop in U, otherwise we would have $(\partial_y f, \partial_z f) = 0$ (and thus $\mathbf{t}_x = 0$) in $U \subset B$.

We deduce that each connected component of $f(\mathbf{x}) = f_0$ in U intersects ∂U in two points.

Now suppose that Φ_{x_0} is not injective on U, so that we have two points $p_1, p_2 \in U$ such that $\Phi_{x_0}(p_1) = \Phi_{x_0}(p_2)$.

If p_1 and p_2 are on the same connected component of the level set $f(\mathbf{x}) = f_0$ (where $f_0 = f(p_1) = f(p_2)$) in U, then g reaches the same value at p_1 and p_2 on this level set, so that by Role's theorem, there exists a point $p \in U$ between p_1 and p_2, such that $\mathrm{Jac}(\Phi_{x_0})(p) = \mathbf{t}_x(p) = 0$. By hypothesis, this is impossible.

Thus p_1 and p_2 belongs to two different connected components of $f(\mathbf{x}) = f_0$ in U. Consequently the value f_0 is reached at 4 distinct points of ∂U, which implies that f has at least 4 extrema on ∂U.

Now note that up to a change of variable $z = a_2 - z$, we can assume that $\partial_z f > 0$ on both $y = a_1, y = b_1$ faces. Then if $\partial_y f < 0$ on $z = a_2$, we have $f(x_0, a_1, b_2) > f(x_0, a_1, a_2) > f(x_0, b_1, a_2)$ and (a_2, a_3) is not a local extremum. Otherwise $\partial_y f > 0$ and (b_1, a_2) is not a local extremum. In both cases, we do not have 4 extrema, which proves that σ_{x_0} is injective and that the intersection of $\mathcal{C}$ with the plane $x = x_0$ in B is at most one point. So by proposition 4.1, we deduce that $\mathcal{C}$ is regular in B.

A similar criterion applies by symmetry, exchanging the roles of the x, y, z coordinates. If one of these criteria applies with $\mathbf{t}_i(x) \neq 0$ on B (for $i = x, y, z$), we will say that $\mathcal{C}$ is i-regular on B.

4.2 Tests of Regularity

From a practical point of view, the test $\mathbf{t}_i(x) \neq 0$ or $\partial_i(h) \neq 0$ for $i = x, y$ or z, $h = f$ or g can be replaced by the stronger condition that their coefficients on the Bernstein basis of B have a constant sign, which is straightforward to check. Similarly, such a property on the faces of B is also direct, since the coefficients of a polynomial on a facet form a subset of the coefficients of this polynomial in the box.

In addition to these tests, we also test whether both surfaces penetrate the cell, since a point on the curve must lie on both surfaces. This test could be done by looking at the sign changes of the Bernstein coefficients of the surfaces with respect to that cell. If no sign change occurs, we can rule out the possibility that the cell contains any portion of the curve $\mathcal{C}$, and thus terminate the subdivision early. In this case, we will also say that the cell is regular.

4.3 Algorithm of Connection

This regularity criterion is sufficient for us to uniquely construct the topological graph g of $\mathcal{C}$ within B. Without loss of generality, we suppose that the curve $\mathcal{C}$ is x-regular in B. Hence, there is no singularity of $\mathcal{C}$ in B. Furthermore, this also guarantees that there is no 'turning-back' of the curve tangent along x-direction, so the mapping of $\mathcal{C}$ onto the x-axis is injective. Intuitively, the mapped curve

should be a series of non-overlapping line segments, whose ends correspond to the intersections between the curve $\mathcal{C}$ and the cell, and such a mapping is injective.

This property leads us to a unique way to connect those intersection points, once they are computed in order to obtain a graph representing the topology of $\mathcal{C}$.

Algorithm 4 (Connection for a x-regular space curve.)
INPUT*: a curve $\mathcal{C}$ x-regular in a domain $B = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2]$.*

1. *Compute the points $p_1, \ldots, p_k$ of $\mathcal{C} \cap \partial B$.*
2. *Remove the points on the edges of B for which there is no interior tangent vector of $\mathcal{C}$ to B. Duplicate the points where the curve is tangent to a face of B and on the same side of this facet at this point.*
3. *Sort this set of points by lexicographic order with $x > y > z$.*
4. *Connect by arcs the pairs of consecutive odd and even index points $(\boldsymbol{v}_{2\,i-1}, \boldsymbol{v}_{2\,i})$, $i = 1 \leq \cdots \leq k/2$, in this sorted list.*

OUTPUT*: a set of points on the curve and arcs connecting them.*

A similar algorithm applies by exchanging x with y or z.

Notice that in order to apply this algorithm, we need to compute the points of $\mathcal{C} \cap B$, that is to solve a bivariate system of each facet of B. This is performed by applying the algorithm described in Section 2.3.

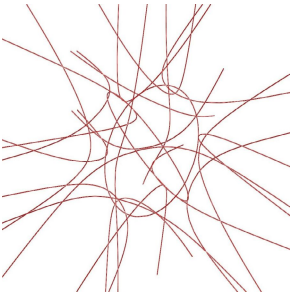
The special treatment of points of $\mathcal{C}$ on an egde of B or where $\mathcal{C}$ is tangent to a face requires the computation of tangency informations at these points. This is performed by evaluating the derivatives of the defining equations of $\mathcal{C}$ at these points.

4.4 Algorithm of Subdivision

Collecting these properties, we arrive at the following subdivision algorithm, which subdivides the domain B until some size ϵ, if the curve is not regular in B.

Algorithm 5 (Topology of a space curve)
INPUT*: a curve $\mathcal{C}$ defined by equations $f_1 = 0, f_2 = 0, \ldots, f_k = 0$ and a domain $B = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^3$ and $\epsilon > 0$.*

- *For $1 \leq i < j \leq k$,*
  - *compute the Bernstein coefficients of the x, y, z coordinates of $\nabla f_i \wedge \nabla f_j$ in B*
  - *check that they are of the same sign for one of the coordinates (say x);*
  - *check that x-regularity condition on the facets of B.*
- *If such a pair (i, j) satisfying the regularity condition exists,*
  - *Compute the points of $\mathcal{C} \cap \partial B$;*
  - *Connect them (Algorithm 4).*



This curve is defined by $f(x,y,z) = 0$, $\partial_z f(x,y,z) = 0$ where

$$f(x,y,z) = 4(\tau^2 x^2 - y^2)(\tau^2 y^2 - z^2)(\tau^2 z^2 - x^2) \\ -(1+2\tau)(x^2+y^2+z^2-1)^2, \tau = \frac{1}{2}(1+\sqrt{5}),$$

of degree 6 is defining the surface S called Barth's Sextic. This curve, called the polar curve of S in the z direction, is of degree $30 = 6 \times 5$. We compute the topology by approximating the coefficients of f and $\partial z f$ by floating point numbers.

Fig. 5. The polar curve of Barth sextic

- *else if $|B| > \epsilon$, subdivide B and proceed recursively on each subdomain.*
- *otherwise find a point p in B°, compute the point $\mathcal{C} \cap \partial B$ and connect them to p.*

OUTPUT*: a set of points p and a set of arcs connecting them.*

As in the 2D case, we have the following 'convergence' property:

Proposition 4.3. *For $\epsilon > 0$ small enough, the graph of points and arcs computed by the algorithm has the same topology as $\mathcal{C} \cap B$.*

5 Surfaces

In this section, we consider a surface $\mathcal{S}$ defined by the equation $f(x,y,z) = 0$, with $f \in \mathbb{Q}[x,y,z]$. We assume that f is squarefree, that is f has no irreducible factors of multiplicity ≥ 2. The results presented in this section extend those of [28] to the treatment of singular surfaces. However, the complexity analysis provided in [28] has not yet been carried out for such surfaces.

Unlike in the 2 dimensional case, the topology of the singular locus and the way to smooth locus is attached to it can be really complicated. To handle such complexity we rely on a technique from singularity theory, namely stratification. For instance, this technique has been fruitfully used in [38] to extend Morse theory to singular varieties. The idea behind stratification is that one can partition a set into submanifolds (the strata) that are connected nicely. Historically, it has been hard to give a formal meaning to 'connect nicely'. The first idea is that the strata should not be transverse to each other.

Unfortunately this straightforward notion of 'nice connection' is not strong enough to ensure good topological properties. A good example is given by the 'double trumpet' (see Fig. 6) picture: all the tangent planes connect smoothly to the singular line, nevertheless one would like to tell that the origin of the two trumpets is special. This is achieved by a stronger condition that was defined by H.Whitney in [39] that is now called condition B.

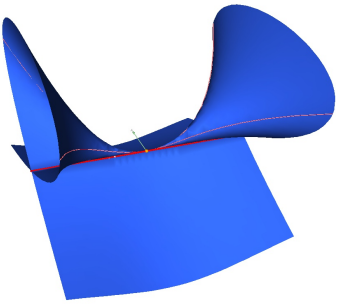


Fig. 6. The double trumpet surface

Definition 5.1 (Whitney's condition B). *$\forall M, N$ submanifolds, $\forall x \in M \bigcap \overline{N}$, we say that (M, N) satisfies condition B at x iff $\forall x_n \in M$ converging to x, $\forall y_n \in N$ also converging to x, such that for some line l and tangent space τ $\lim_{n\to\infty}(x_n y_n) = l$ and $\lim_{n\to\infty} T_{y_n}(N) = \tau$ then $l \subset \tau$ where $(x_n y_n)$ denotes the line passing through those points.*

The main point that Whitney proved is that for algebraic varieties, 'condition B is a stratifying condition'. This means that recursively eliminating those points where condition B is not met for the smooth locus, and relegating them to a new subvariety on which we will once again eliminate points not satisfying condition B in the subvariety, and so on... is a process that actually yields a partition of the variety into submanifolds where all pairs satisfy Whitney's condition B.

Definition 5.2 (Whitney stratification). *Let E a set, and S a partition of E, $\mathcal{S}$ is Whitney stratification of E iff*

- *every stratum $\sigma \in \mathcal{S}$ is a submanifold.*
- *$\forall \sigma_1, \sigma_2 \in \mathcal{S}$, $(\sigma_1 \bigcap \overline{\sigma_2} \neq \emptyset) \Rightarrow \sigma_1 \subset \overline{\sigma_2}$. This is called the boundary condition.*
- *$\forall K \subset \mathbb{R}^n$ compact, $\{\sigma \in \mathcal{S} : \sigma \bigcap K \neq \emptyset\}$ is finite. This is the local finiteness condition.*
- *$\forall \sigma_1, \sigma_2 \in \mathcal{S}$, $\forall x \in (\sigma_1 \bigcap \overline{\sigma_2})$, (σ_1, σ_2) satisfies condition B at x*

Actually condition B implies the boundary condition. But because this is not straightforward it is usual to include the boundary condition in the definition as it helps a lot imagining how the different strata are connected together. Another such geometric result is that if $\sigma_1 \subset \partial\sigma_2$ then $\dim(\sigma_1) < \dim(\sigma_2)$.

For surfaces we have an effective way to determine such a Whitney stratification. It is based on a 3d-curve called 'polar variety' that is made of the critical points of a linear projection to a plane, in particular it contains the singular locus.

- The 2-dimensional strata is the smooth part of the surface minus the polar curve.
- The 1-dimensional strata is the smooth part of the polar variety.
- The 0-dimensional strata is the singular points of the polar variety.

The proof that this is indeed a Whitney stratification has been carried out in [20] and it is a consequence of a theorem of Speder [40].

The main result that Whitney stratifications yield is topological triviality. Going back to the 'double trumpet' case, the topology of the intersection of the surface with a vertical plane changes when the plane reaches the central point. That point is not distinguished by condition A because the trumpet narrows fast, unlike a cone $x^2 + y^2 - z^2$ for instance. From this simple geometric fact that the topology changes, one can tell that this central point has to be singled out by condition B. This link between Whitney stratifications and topological triviality has been formalized by Thom and is known as Thom's isotopy lemma ([41,42]).

Definition 5.3 (Proper stratified submersion)
Let $f : \mathbb{R}^n \to \mathbb{R}^m$, and a set $Z \subset \mathbb{R}^n$ Whitney stratified by $\mathcal{S}$, we say that f is a proper stratified submersion for $\mathcal{S}$ iff $f|_Z$ proper and $\forall \sigma \in \mathcal{S}$, the differential map $\partial f|_\sigma$ has constant rank m.

Theorem 5.1 (Thom's isotopy lemma)
$\forall Z \subset \mathbb{R}^n$, $\forall \mathcal{S}$ Whitney stratification of Z, $\forall g : \mathbb{R}^n \to \mathbb{R}^m$ such that g smooth proper stratified submersion for $\mathcal{S}$,
$\exists h : Z \to \mathbb{R}^m \times (g^{-1}(0) \bigcap Z)$ stratum preserving homeomorphism, such that h smooth on each stratum, and $\Pi_{\mathbb{R}^m} \circ h = g$.

Now that we have acquired a proper understanding of how an algebraic variety can be decomposed into smooth strata that fit nicely together, we can actually take advantage of this decomposition to determine conditions sufficient to know the topology in small enough boxes. Indeed, as our algorithm proceeds by subdivision, we will finally end up with boxes as small as we want. So we just have to take care of what happens locally. We know we can split the surface into patches of 2-dimensional strata (the smooth part), 1-dimensional strata (singular curves), and 0-dimensional strata (singular points were the 1-dimensional strata do not meet condition B). Topologically we can characterize the topological situation as follows:

- Near a 2-dimensional stratum the topology is the same as a hyperplane.
- Near a 1-dimensional stratum the topology is the same as a cylinder on a singular planar curve.
- Near a 0-dimensional stratum the topology is the same as a cone with center, the isolated point of one of these 0-dimensional stratum and with base, the intersection of the surface with a small sphere and

And we know only one of these three situations can and will happen locally. So we just have to design a solution for each one of the above three cases. For efficiency reasons the criteria we have designed work for situations more general than the limit three cases. The 2-dimensional strata criterion can succeed even with several hyperplanes in the box not only just one. For the 1-dimensional stata we can triangulate even when some patches of the 2-dimensional strata lie in the

box even though they are disconnected from the singular locus (in the box). In the case of the 0-dimensional strata we need to have the exact topology in the box. Of course the criteria eventually succeed if the box is small enough. We now describe each one of these criteria and the matching connection algorithm.

5.1 Regularity Criterion for Surfaces

In this section, we consider a domain $B = [a, b] \times [c, d] \times [e, f] \subset \mathbb{R}^3$ and a surface $\mathcal{S} \subset B$. The x-faces (resp. y, z-facet) of B are the planar domains of the boundary ∂B of B, orthogonal to the direction x (resp. y, z).

Definition 5.4. *The surface $\mathcal{S}$ is z-regular (resp. y, z-regular) in the domain B if,*

- *$\mathcal{S}$ has no tangent line parallel to the z-direction (reps. x, y-direction),*
- *$\mathcal{S} \cap F$ is regular, for F a z-facet (resp. x, y-facet) of B.*

We will say that $\mathcal{S}$ is *regular* in B if it is regular in B for the direction x, y or z. Here again, if a point $p \in \mathcal{S}$ is singular, then any line through this point is tangent to $\mathcal{S}$ at p. Thus a surface in B which is regular is also smooth.

Proposition 5.1. *If $\mathcal{S}$ is regular, then its topology is uniquely determined by its intersection with the edges of B.*

Proof. Exchanging the role of x, y or z if necessary, we can assume that $\mathcal{S}$ is z-regular in $\mathcal{S}$. As $\mathcal{S}$ has no vertical tangent in B, there is no closed connected component of $\mathcal{S}$ in B and each patch of $\mathcal{S} \cap B$ intersects the boundary ∂B.

As there is no point of $\mathcal{S}$ with vertical tangent in B, $\mathcal{C}_F = \mathcal{S} \cap F$ is z-regular for any x or y facet F of B. By hypothesis, the curve $\mathcal{C}_F = \mathcal{S} \cap F$ is also regular for the z-facets of B. By proposition 3.1, for any facet F of B, the topology of $\mathcal{C}_F$ is uniquely determined by the points of $\mathcal{C}_F \cap \partial F$. In other words, the topology of $\mathcal{S} \cap \partial B$ is uniquely determined by the intersection of $\mathcal{S}$ with the edges of B.

Since $\partial\mathcal{S} \cap B \subset \partial B$, otherwise $\mathcal{S}$ would not be smooth in B° or would have a vertical tangent, each patch $\mathcal{P}$ of $\mathcal{S} \cap B$ is simply connected. Moreover, the boundary $\partial\mathcal{P}$ is made of branches of $\mathcal{C}_F$ for F a facet of B, which share an end point on the edges of B. Thus the patches $\mathcal{P}$ of $\S \cap B$ are uniquely determined by the cycles of arcs of $\mathcal{S} \cap \partial B$.

This leads to the following algorithm:

Algorithm 6 (Connection of a regular surface)
INPUT*: A domain $B \subset \mathbb{R}^3$ and a surface $\mathcal{S}$ which is regular in B.*

- *Compute the set Ξ of intersection points of $\mathcal{S}$ with the edges of B;*
- *Deduce the set Σ of branches of $\mathcal{S} \cap \partial B$ (using algorithm 2);*
- *Extract the connected components $\mathcal{P}_1, \ldots, \mathcal{P}_k$ of the graph $\Gamma = (\Xi, \Sigma)$ with vertices Ξ and edges Σ.*

OUTPUT*: The boundary of each patch of $\mathcal{S} \cap B$ described by the cycle of branches $\mathcal{P}_i$, for $i = 1, \ldots, k$.*

Since these arcs form the boundary of $\mathcal{S} \cap B$, locally the part of the surface which is inside B is on "one side" of this arc. This defines a half plane on one side on the tangent line on the tangent plane at a point of such arc.

For a point p on an edge of B, which belongs to two arcs, we define the interior tangent sector of $\mathcal{S}$ at p as the intersection of the two corresponding half planes and denote it by $T_p^i(S)$.

As in the 2D case, simple tests of regularity can be derived from the representation of f in the Bernstein basis.

Lemma 5.1. *Let (u, v, w) be any permutation of (x, y, z). Suppose that $\partial_u f \neq 0$ in B and that $\mathcal{S} \cap F$ is regular on the u-facets F of B. Then S is regular in B.*

Proof. As $\partial_u f \neq 0$ in B, $\mathcal{S}$ has no tangent parallel to the u-direction in B and its u-facets are regular, so that $\mathcal{S}$ is u-regular in B.

Proposition 5.2. *Let (u, v, w) be any permutation of (x, y, z). Suppose that the coefficients of $\partial_u f$ in the Bernstein basis of B have the same sign $\in \{-1, 1\}$ and that the coefficients of $\partial_v f$ or $\partial_w f$ on the u-facets of B are also of the same sign $\in \{-1, 1\}$. Then $\mathcal{C}$ is regular in B.*

Proof. By the convex hull property $\partial_u f \neq 0$ in B and $\partial_v f \neq 0$ or $\partial_w f \neq 0$ on a u-facet F of B. This implies that the curve $\mathcal{S} \cap F$ is regular in F and therefore that $\mathcal{S}$ is regular in B.

This criterion implies that in the valid cells, the derivative of f in one direction is of constant sign and on the two faces transversal to this direction, another derivative is of constant sign. This may be difficult to obtain, when a point of the surface where two derivatives vanish is on (or near) the boundary of the cell. A situation where $\partial_u f \neq 0$ but where both derivative $\partial_v f$, $\partial_w f$ are not of constant sign on a u-facet F of B, can be handled by applying recursively the 2D algorithm of the facet F.

5.2 The Polar Variety

As we said before, the way we get a handle on the lower dimensional strata is the polar variety. This object defines a 3d-curve that we will consider to be the closure of our 1-dimensional strata. And we will define the 0-dimensional strata as the singular points of this curve. It was proved in [20] as a consequence of a theorem of Speder [40] that this process yields a Whitney stratification.

Now we explain how this object is defined and computed. The polar variety is dependent on a direction that can be chosen arbitrarily. For simplicity, we will consider that this direction is the z-direction.

Definition 5.5. *The polar variety of $\mathcal{S}$ for the direction z, denoted $\mathfrak{C}_z(\mathcal{S})$ is the set of points $p \in \mathcal{S}$, such that the line at p of direction z is tangent to $\mathcal{S}$. It is defined by the equations $f(x, y, z) = 0$, $\partial_z f(x, y, z) = 0$.*

Notice that $\mathfrak{C}_z(\mathcal{S})$ contains the set of singular points of $\mathcal{S}$ (satisfying $f = \partial_x f = \partial_y f = \partial_z f = 0$).

As we assume that f is squarefree, if $\partial_z f \neq 0$ then the algebraic variety defined by $f = 0, \partial_z f = 0$ is a curve (ie. of dimension 1). However, it can contain irreducible components with multiplicities ≥ 2, which might induce problems in subdivision techniques.

In order to remove these multiplicities, we need to compute the radical $I(\mathfrak{C}_z(\mathcal{S}))$ of the ideal $I := (f, \partial_z f)$.

Let us denote by $J(f, g)$ the ideal of $\mathbb{R}[x, y, z]$ generated by the coordinates of $\nabla f \wedge \nabla g$. Recall that for two ideal I, I' of $\mathbb{R}[x, y, z]$, $(I : I') = \{f \in \mathbb{R}[x, y, z]; \forall g \in I', f\, g \in I\}$. We compute the radical ideal of I, as follows:

Proposition 5.3. *If $\partial f \neq 0$, the radical $I(\mathfrak{C}_z(\mathcal{S}))$ is $(f, \partial f) : J(f, \partial_z f)$.*

Proof. It is a direction application of [43].

This ideal division $(I : J(I))$ can be computed by classical algebraic techniques, such as Gröbner bases [44]. We consider this step as a preprocessing step, which computes a minimal set of generators $g_1, \ldots, g_k$: $I(\mathfrak{C}_z(\mathcal{S})) = (g_1, \ldots, g_k)$. Hereafter, we are going to assume that these generators are computed before the subdivision process starts.

We are going to use the topology of the polar curve in order to deduce the topology of the whole surface $\mathcal{S}$. For that purpose, we use algorithm 5.

Let us describe now how the topology of the surface is deduced in a domain where the polar curve is x-regular (we handle the other direction symmetrically). We will suppose moreover that this polar curve is connected in B, that is made of one x-regular arc connecting two points on $\mathcal{S} \cap \partial B$ with distinct x-coordinates. We also suppose that the topology of $\mathcal{S} \cap \partial B$ is known, that for each facet F a set of arcs of F connecting points on the edges or on the polar curve are known. For that purpose, we apply the algorithm described in section 3. This all happens (in one direction x or y or z) when we are close enough to the 1-dimensional strata but away from the 0-dimensional strata.

Because the polar curve is x-regular we know that the projection to the x-axis is a stratified submersion. It is proper because we look at what's in the box. Therefore by Thom's isotopy lemma we know that we have a trivializing homeomorphism, which in this case just deforms the surface in the box to a cylinder over a planar curve (that can be obtained by any intersection of the surface with a $x = constant$ plane).

In order to recover those patches of the surface that connect along the polar curve, we follow the arcs on the facets, starting from a point on the polar variety. The remaining arcs which are not involved in these patches are connected using the algorithm 6 for regular surfaces. This leads to the following algorithm:

Algorithm 7 (Connection for surface with regular polar variety)
INPUT: *A surface $\mathcal{S}$ and a box B where the polar variety of $\mathcal{S}$ is regular and connected. On each facet, a set of arcs connecting points on the boundary, or on the polar curve, describing the topology of $\mathcal{S} \cap \partial B$.*

- *Remove the arc of the polar variety from the initial set of arcs and its two points on ∂B.*

- *Compute the connected components of the remaining set of arcs and points.*
- *For each connected component which starts and ends at the two points of the polar variety, add the polar arc and form the corresponding patch.*
- *For each of the remaining connected components, which form cycles, build the corresponding patches.*

OUTPUT*: The boundary of each patch of* $\mathcal{S} \cap B$ *described as a list of cycles of arcs, taken from the input set of arcs.*

5.3 The Singular Locus of the Polar Curve

Finally, we compute the topology of the surface around the 0-dimensional strata, that is the singular points of the curve. Indeed, as we have a radical ideal defining the curve, they can be obtained by looking at the points where all the 2x2 determinants of the derivatives of the generators of the ideal $I(\mathfrak{C}_z(\mathcal{S}))$ vanish.

We are now left with those boxes where the polar variety is singular. We can assume that there is only one singular point of the polar variety in the box, and that the variety is close to its tangent cone. Because we are close to the tangent cone, all the tangent lines and planes to the 1 and 2-dimensional strata almost go through the singular point of the polar variety. Therefore they are transverse to the balls containing the singular point and we can apply Thom's isotopy lemma to deduce that the topology in the balls is invariant by radius reduction. Thus the topology in a ball is the same as a cone with center the singular point of the polar curve, and with base the intersection of the surface with the boundary of the ball. For a small enough box B, which contains the singular point p, the surface will also be like cone with center p and with base $\mathcal{S} \cap \partial B$.

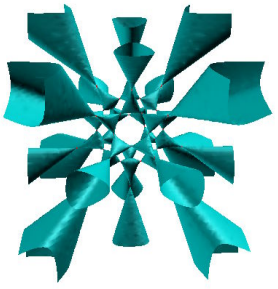
In order to determine the topology of the surface $\mathcal{S}$ in B, we compute the topology of $\mathcal{S} \cap \partial B$ using Algorithm 3. Then we just connect all points and arcs in the boundary to the central point.

Algorithm 8 (Topology of a surface)
INPUT*: a surface* $\mathcal{S}$ *defined by a squarefree equation* $f(x, y, z) = 0$*, a domain* $B_0 = [a_0, b_0] \times [a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^3$ *and* $\epsilon > 0$*.*

- *Compute generators* $g_1, \ldots, g_k$ *of the ideal* $I(\mathfrak{C}_z(\mathcal{S})) = (f, \partial_z f) : J(f, \partial_z f)$*.*
- *Compute the Bernstein coefficients of the* f *and* g_i *in the Bernstein basis of* $B := B_0$*.*
- *If* $\mathcal{S}$ *is regular in* B*, compute its topological structure by Algorithm 6.*
- *Else if the polar variety* $\mathfrak{C}_z(\mathcal{S})$ *is regular and connected in* B*, compute the topological structure of* $\mathcal{S} \cap \partial B$ *by Algorithm 3 on each facet of* B
- *Else if* $|B| > \epsilon$*, subdivide the box* B *and proceed recursively on each subdomain.*
- *Otherwise find a point* p *in* B°*, compute the topological structure of* $\mathcal{S} \cap \partial B$ *by Algorithm 3 and its link over* p*.*

OUTPUT*: a set of points, arcs and patches and adjacency relations describing the topology of* $\mathcal{S} \cap B_0$ *arcs connecting them.*

As in the previous case, for $\epsilon > 0$ small enough, the output of this algorithm is topologically equivalent to $\mathcal{S} \cap B$.



Here is the Barth sextic surface whose polar variety has been computed in and shown in Fig. 5. This surface of degree 6 has the maximun number of isolated singularities for this degree, that is 65. These singular points are also singular points of its polar variety.

Fig. 7. Barth sextic

6 Conclusion

We have presented a comprehensive set of algorithms to determine the topology of singular curves both planar and 3-dimensional, as well as singular surfaces. The approach we chose is subdivision, it allows us to handle problems locally and to make use of numerical approximations and still test properties exactly.

Conceptually, the stability needed to be able to do numerical rounding and still get a correct result is achieved through transversality. Indeed, if two objects are transverse, they will still be transverse after a small perturbation. When striving for this goal our enemy is multiplicity. We take care of it by stratifying the variety and eliminating multiplicity in codimension 1.

Stratifying is easy for curves, but we rely on a theorem of Speder to do this for surfaces. This is the main tumble stone when trying to generalize this method to higher dimensions as then it is not clear how to get a Whitney stratification efficiently.

Getting rid of multiplicity in codimension 1 is nothing more than computing the radical of an ideal. It is easy for principal ideals, this is 'taking the squarefree part of the generator'. It is trickier to do it efficiently in general, but we managed to do it in our case (when computing the polar variety).

Finally these algorithms still lack a precise assessment of their complexity. Although we know from experience that they are fast. The situation is understood as we can relate the size of the boxes to a measure of transversality. The thorough analysis would have been to heavy for this article as it requires lengthy developments on measures of transversality and bounds on polynomials.

Acknowledgements

The topology computation and visualization of this paper were performed by the software:

AXEL (`http://www-sop.inria.fr/galaad/axel/`)
SYNAPS (`http://www-sop.inria.fr/galaad/synaps/`).

This work is partially supported by AIM@Shape (IST Network of Excellence 506766) and ACS (IST FET Open 006413).

References

1. Bloomenthal, J.: Introduction to implicit surfaces. Morgan Kaufmann, San Francisco (1997)
2. Grandine, T.A., Klein, F.W.: A new approach to the surface intersection problem. Computer Aided Geometric Design 14, 111–134 (1997)
3. Farin, G.: An ssi bibliography. In: Geometry Processing for Design and Manufacturing, pp. 205–207. SIAM, Philadelphia (1992)
4. Kalra, D., Barr, A.: Guaranteed ray intersections with implicit surfaces. In: Proc. of SIGGRAPH, vol. 23, pp. 297–306 (1989)
5. Lorensen, W., Cline, H.: Marching cubes: a high resolution 3d surface construction algorithm. Comput. Graph. 21(4), 163–170 (1987)
6. Zhou, Y., Baoquan, C., Kaufman, A.: Multiresolution tetrahedral framework for visualizing regular volume data. In: Proc. of Visualization '97, pp. 135–142 (1997)
7. Bloomenthal, J.: Polygonization of implicit surfaces. Computer-Aided Geometric Design 5(4), 341–355 (1988)
8. Bloomenthal, J.: An implicit surface polygonizer. In: Heckbert, P. (ed.) Graphics Gems IV, pp. 324–349. Academic Press, Boston, MA (1994)
9. Hartmann, E.: A marching method for the triangulation of surfaces. Visual Computer 14(3), 95–108 (1998)
10. Witkin, A., Heckbert, P.: Using particles to sample and control implicit surface. In: Proc. of SIGGRAPH, pp. 269–277 (1994)
11. Boissonnat, J.D., Oudot, S.: Provably good sampling and meshing of surfaces. Graphical Models 67, 405–451 (2005)
12. Dey, T.K.: Curve and Surface Reconstruction: Algorithms with Mathematical Analysis (Cambridge Monographs on Applied and Computational Mathematics). Cambridge University Press, Cambridge (2006)
13. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Brakhage, H. (ed.) Automata Theory and Formal Languages. LNCS, vol. 33, pp. 134–183. Springer, Heidelberg (1975)
14. Keyser, J., Culver, T., Manocha, D., Krishnan, S.: Efficient and exact manipulation of algebraic points and curves. Computer-Aided Design 32(11), 649–662 (2000)
15. González-Vega, L., Necula, I.: Efficient topology determination of implicitly defined algebraic plane curves. Comput. Aided Geom. Design 19(9), 719–743 (2002)
16. Gatellier, G., Labrouzy, A., Mourrain, B., Técourt, J.P.: Computing the topology of 3-dimensional algebraic curves. In: Computational Methods for Algebraic Spline Surfaces. LNCS, pp. 27–44. Springer, Heidelberg (2005)
17. Alcázar, J.G., Sendra, J.R.: Computing the topology of real algebraic space curves. J. Symbolic Comput. 39, 719–744 (2005)
18. Fortuna, E., Gianni, P., Parenti, P., Traverso, C.: Computing the topology of real algebraic surfaces. In: ISSAC '02. Proceedings of the 2002 international symposium on Symbolic and algebraic computation, pp. 92–100. ACM Press, New York (2002)
19. Cheng, J.S., Gao, X.S., Li, M.: Determining the topology of real algebraic surfaces. In: Martin, R., Bez, H., Sabin, M.A. (eds.) Mathematics of Surfaces XI. LNCS, vol. 3604, pp. 121–146. Springer, Heidelberg (2005)
20. Mourrain, B., Técourt, J.: Isotopic meshing of a real algebraic surface. Technical Report 5508, INRIA Sophia-Antipolis (2005)

21. Sherbrooke, E.C., Patrikalakis, N.M.: Computation of the solutions of nonlinear polynomial systems. Comput. Aided Geom. Design 10(5), 379–405 (1993)
22. Elber, G., Kim, M.S.: Geometric constraint solver using multivariate rational spline functions. In: Proc. of 6th ACM Symposium on Solid Modelling and Applications, pp. 1–10. ACM Press, New York (2001)
23. Mourrain, B., Pavone, J.P.: Subdivision methods for solving polynomial equations. Technical Report 5658, INRIA Sophia-Antipolis (2005)
24. Sederberg, T.: Algorithm for algebraic curve intersection. Computer Aided Design 21, 547–554 (1989)
25. Hass, J., Farouki, R.T., Han, C.Y., Song, X., Sederberg, T.W.: Guaranteed Consistency of Surface Intersections and Trimmed Surfaces Using a Coupled Topology Resolution and Domain Decomposition Scheme. Advances in Computational Mathematics (to appear)
26. Seong, J.K., Elber, G., Kim, M.S.: Contouring 1- and 2-Manifolds in Arbitrary Dimensions. In: SMI'05, pp. 218–227 (2005)
27. Liang, C., Mourrain, B., Pavone, J.: Subdivision methods for 2d and 3d implicit curves. In: Computational Methods for Algebraic Spline Surfaces, pp. 171–186. Springer, Heidelberg (to appear) also available at `http://hal.inria.fr/inria-00130216`
28. Alberti, L., Comte, G., Mourrain, B.: Meshing implicit algebraic surfaces: the smooth case. In: Maehlen, M., Morken, K., Schumaker, L.L. (eds.) Mathematical Methods for Curves and Surfaces: Tromso'04, Nashboro, pp. 11–26 (2005)
29. Farin, G.: Curves and surfaces for computer aided geometric design: a practical guide. Comp. science and sci. computing. Academic Press, Boston, MA (1990)
30. Emiris, I.Z., Mourrain, B., Tsigaridas, E.P.: Real algebraic numbers: Complexity analysis and experimentations. In: Reliable Implementation of Real Number Algorithms: Theory and Practice. LNCS, Springer, Heidelberg (to appear) also available at `http://hal.inria.fr/inria-00071370`
31. Eigenwillig, A., Sharma, V., Yap, C.K.: Almost tight recursion tree bounds for the descartes method. In: ISSAC '06. Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation, pp. 71–78. ACM Press, New York (2006)
32. Pavone, J.: Auto-intersection de surfaces pamatrées réelles. PhD thesis, Université de Nice Sophia-Antipolis (2004)
33. Floater, M.S.: On zero curves of bivariate polynomials. Journal Advances in Computational Mathematics 5(1), 399–415 (1996)
34. Abhyankar, S.: Algebraic Geometry for Scientists and Engineers. American Mathematical Society, Providence, R.I. (1990)
35. Walker, R.: Algebraic curves. Springer, Heidelberg (1978)
36. Lloyd, N.G.: Degree theory. Cambridge Tracts in Mathematics, Cambridge University Press, Cambridge No. 73 (1978)
37. Dickenstein, A., Rojas, M.K.R., Shihx, J.: Extremal real algebraic geometry and a-discriminants (preprint)
38. Goresky, R.M.M.: Stratified Morse Theory. Springer, Heidelberg (1988)
39. Whitney, H.: Elementary structure of real algebraic varieties. Annals of Math. 66(2) (1957)
40. Speder, J.P.: Équisingularité et conditions de Whitney. Amer. J. Math. 97(3) (1975)
41. Thom, R.: Ensembles et morphismes stratifiés. Bull. Amer. Math. Soc. 75 (1969)

42. Mather, J.: Notes on topological stability. Harvard University (1970)
43. Eisenbud, D., Huneke, C., Vasconcelos, W.: Direct methods for primary decomposition. Invent. Math. 110, 207–235 (1992)
44. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Undergraduate Texts in Mathematics. Springer, Heidelberg (1992)

Quadrangle Surface Tiling Through Contouring

Pierre Alliez

INRIA Sophia-Antipolis, France
pierre.alliez@sophia.inria.fr

Abstract. Our algorithm computes two piecewise smooth harmonic scalar functions, whose isolines tile the input surface into well-shaped quadrangles, without any T-junctions. Our main contribution is an extension of the discrete Laplace operator which encompasses several types of line singularities. The resulting two discrete differential 1-forms are either regular, opposite or switched along the singularity graph edges. We show that this modification guarantees the continuity of the union of isolines across the lines, while the locations of the isolines themselves depend on the global solution to the modified Laplace equation over the whole surface.

1 Introduction

Partitioning a surface into quadrilateral regions is a common requirement in computer graphics, computer aided geometric design and reverse engineering. Such quad tilings are amenable to a variety of subsequent applications due to their tensor-product nature, such as B-spline fitting, simulation with finite elements or finite differences, texture atlasing, and addition of highly detailed modulation maps. Quad meshes are particularly useful in modeling as they aptly capture the symmetries of natural or man-made geometry, allowing artists to design simple surfaces using a quite intuitive placement of quad elements. Automatically converting a triangulated surface (issued from a 3D scanner for instance) into a quad mesh is, however, challenging. Stringent topological conditions make quadrangulating a domain or a surface a rather constrained and global problem [Ede00]. Application-dependent meshing requirements such as edge orthogonality, alignment of the elements with the geometry, sizing, and mesh regularity add further hurdles. We show how to use discrete harmonic forms to solve for two piecewise-smooth scalar fields such that their respective isocontours create a mesh with well-shaped quadrangles at geometrically pertinent edge locations (see Figure 1).

1.1 Previous Work

Due to their wide appeal in various communities, quad meshes have been the subject of a large number of papers presenting different algorithms for the generation of isotropic or anisotropic quad elements. Comprehensive reviews, found for instance in [ACSD+03, BMRJ04, AUGA05, DKG05], hint at a need for algorithms offering more *control* on the mesh regularity, as well as on the shape,

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 29–41, 2007.
© Springer-Verlag Berlin Heidelberg 2007

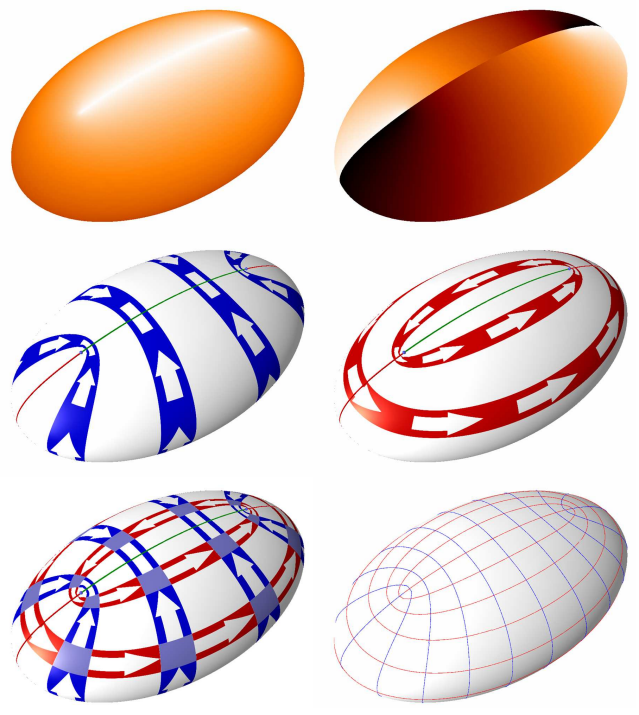


Fig. 1. Overview of our technique. Top: two harmonic potentials. Middle: pair of 1-forms associated to the potentials. Bottom right: an isocontouring of these potentials results in a quadrangle tiling.

size and alignment of the mesh elements with geometric or semantic features. A clustering-based method presented in [BMRJ04] manages to limit the number of extraordinary vertices in the final mesh, without guaranteeing the location of singularities. A recent Morse-theoretic approach in [DBG+06] provides improved results, but the modified relaxation [KLS03] involved in this method still allows no control over design, resulting in singularities at conspicuous places and elements of arbitrary shapes. Another technique allows fully regular quadrilateral meshes (except along a seam) through the use of holomorphic discrete 1-forms [GY03]. Unfortunately, this holomorphic requirement leaves little control over the local alignment of the mesh elements and creates potentially large area distortion, even after optimization [JWYG04]. A recent technique proposes a radically different approach to conformal parameterization with *arbitrary* cone singularities [KSS06]: distortion is concentrated at carefully chosen places so as to allow better, global control of area distortion and hence over mesh sizing. Unfortunately, this non-linear method cannot directly control alignment with features, and/or guarantee proper matching of quads through patch boundaries.

A recent trend towards a better control of alignment focuses on vector field topology. A first approach proposed in [ACSD+03, MK04] consists in tracing *curvature lines*, thereby enforcing proper alignment of the mesh edges while creating

a natural quad-dominant network. The placement of these lines are based on local decisions, resulting in numerous hanging lines all over the mesh: T-junctions and poor regularity of the mesh result from this greedy line selection. When targeting higher mesh regularity, a better approach defines these lines as isolevels and steepest descents of a *global* potential [DKG05]. As a result of this type of contouring, the lines are either closed curves (so-called isoparametric flow lines), or streamlines (gradient flow lines) obtained by numerical integration, leading to less T-junctions and irregularity on the final quad mesh. This method allows some design control through user-defined selection of a number of local extrema of the potential (be they points, or even polylines). However, each local extremum corresponds to an index 1 in the gradient field of the potential; because of the Hopf-Poincaré index theorem, this means that a number of *other* singularities (most likely saddles, of index -1) will be consequently created too, as the indices of all singularities of the vector field must sum up to the Euler characteristic of the surface. Therefore, design control does not scale nicely as each additional constraint increases shape distortion of the tiles on the rest of the surface.

1.2 Local Quadrangulation as Contouring

We start by using a "reverse-engineering" argument. Suppose that we *already* have a small surface patch composed of locally "nice" quadrangles, the notion of nice being highly application-dependent. From this mesh, we can first set a local (u, v) coordinate system (with directions e_u and e_v) of the surface to be aligned with the edges of the quads. We can then define a metric $\langle\,,\rangle$ such that $\langle e_u, e_v\rangle$=0 everywhere, and such that lengths of each quad edge are unit. Thus, the mesh is locally defined by integer u- *and* v- isovalues.

In addition the gradients of the two parameters ∇u and ∇v are *orthogonal* in the prescribed metric. The way we have defined the metric also guarantees that we must have the magnitudes of the gradients equal to each other. The two conditions together are known as the Cauchy-Riemann equations for the parameters u and v of this patch:

$$\langle \nabla u, \nabla v\rangle = 0 \qquad \text{and} \qquad \langle \nabla u, \nabla u\rangle = \langle \nabla v, \nabla v\rangle.$$

These two equations can be formulated using the differentials of u and v, as well as the Hodge star induced by our metric. Indeed, the two 0-forms u and v simply satisfy:

$$du = \star dv$$

Notice that we can deduce (by applying d and $d\star$ to the previous equation) that $d \star dv = d \star du = 0$, hence du and dv are both coclosed. Since $d \circ d = 0$, both are also closed. Therefore, du and dv must be *harmonic*. In more traditional notation, both gradient fields are curl- and divergence-free. Another consequence of the coclosedness of the two differentials is that both u and v are *also harmonic*, i.e., their Laplacian vanishes. These properties explain the popularity of harmonic functions in Euclidean space, where orthogonality means $\pi/2$ angles, hence leading to well-shaped quads [DKG05, GY03]. We will also stick to the Euclidean metric for now, to keep our explanations simple.

1.3 Towards Global Contouring

To extend the basic principle explained in the previous section from a local quad mesh to a global quad mesh, one needs to overcome a number of issues.

Necessity of discontinuities. First, globally continuous harmonic scalar potentials are too restrictive for quad meshing purposes. In fact, for the frequent case of a genus-0 closed manifold , there are *no* globally continuous harmonic potentials other than the constant ones, of little worth. A classical way to deal with this problem is to add pole singularities, which amounts to piercing little holes at various locations on the surface. For example: pierce a sphere once at the top and once at the bottom; what remains is a globally continuous harmonic potential u, with extrema at the two poles, thus with flow lines defining longitudes. However, the corresponding v potential cannot be globally continuous since its derivative dv has closed flow lines, namely latitudes. Therefore, the only hope to truly extend the contouring approach is to allow the potentials to be *piecewise continuous*, i.e., only continuous inside non-overlapping patches of the manifold—akin to the traditional notion of charts [GH95, YZ04, KLS03]. We may find a potential v that is continuous everywhere except on a line joining the two poles, on which the jump of the v-value equals a constant. Note that in this case v is discontinuous but dv is not. However, we will see that continuous harmonic 1-forms are not sufficient in general. The reason is that they can only model singularities with *integer* indices, that is poles and saddles. As Figure 2 depicts, these types of singularities create significant distortion in the quadrangulation. To be able to generate fractional singularities, one needs to allow for certain types of discontinuities on 1-forms.

Compatibility conditions. Acknowledging the lack of global continuity, we now assume that the 0-forms (the potentials u and v) can contain *singularities*, i.e., jumps along certain edges. Similarly, their differentials (the 1-forms du and dv, akin to the gradients of each potential) may have singularities at the same locations, i.e., the vector fields representing these 1-forms may jump across patch boundaries. We will denote such a boundary between two continuous patches a singularity line. Fortunately, we will be able to set *simple compatibility conditions* on the jumps of potentials and of their differentials that will guarantee that a global contouring of u and v results in a proper tiling. More precisely, linear constraints of continuity on the two potentials can trivially ensure *continuity of the isolines*: indeed, if we trace all isolines with integer values, then the necessary and sufficient compatibility conditions are that the jumps of the potentials should be integer. On the other hand, the *smoothness of the isolines* will be ensured by a (tweaked, yet still linear) condition of "harmonicity" of the two potentials at patch boundaries. This last condition is, in fact, a continuity condition for 1-forms across the singularity line. We thus call this condition *singular continuity* to convey the notion of smoothness *modulo* the presence of a singularity.

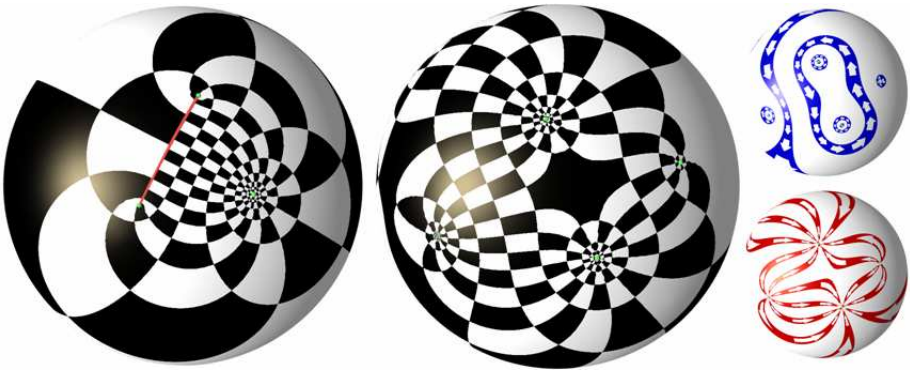


Fig. 2. Undesirable Singularities. Left: a pole (in green) induces too much distortion while a cut (in red) creates T-junctions in the final tiling. Middle and right: more than two poles (of index 1) on a genus-0 surface inevitably create singularities with negative index (saddles), creating large and distorted n-gons.

Singular Continuity of Discrete Forms. As mentioned above, obtaining a quad-dominant tiling on a disk-like patch through contouring two 0-forms u and v is rather easy. However, enforcing a proper tiling throughout the surface requires strong compatibility conditions at each singular line. Fortunately, only three different types of singular continuity across two neighboring patches can happen: *regular* (when both u and v directions individually match between the two patches), *reverse* (when both u and v directions change their orientations across the boundary), and *switch* (when the u and v directions are switched on the shared boundary). Only then can we get a globally consistent tiling of the surface.

1.4 Enforcing Singular Continuity of Forms

We now go over the various cases of continuity. As our technique uses two linear equations per vertex, we describe the different vertex types that we can encounter on a mesh: a vertex can be strictly within a patch, or on a particular type of singularity line.

Free Vertices. When a vertex i is within a patch, i.e., not on any singularity line, we simply wish to enforce harmonicity of both 0-forms u and v. Consequently, the celebrated harmonicity condition [PP93] is imposed on this vertex, yielding:

$$\boxed{\sum_{j \in \mathcal{N}(i)} w_{ij} \begin{pmatrix} u_i - u_j \\ v_i - v_j \end{pmatrix} = 0}$$

where the index j goes through all the immediate neighboring vertices of i, u_k (resp., v_k) represents the value of u (resp., v) at vertex indexed k. For the

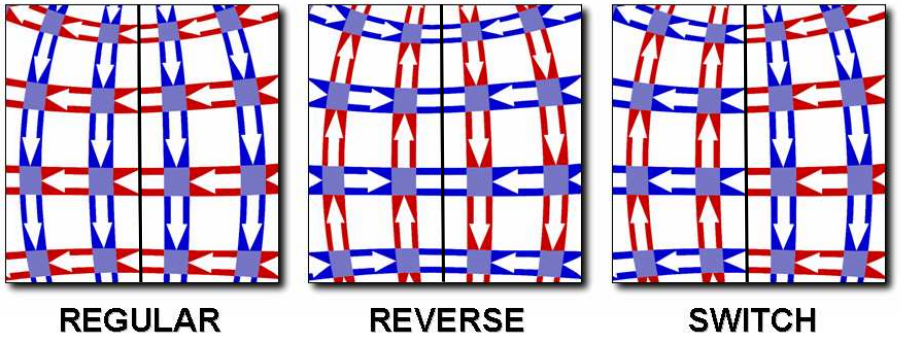


Fig. 3. Singular continuity. Three different types of continuity through a singularity line. Blue/red arrows are along isolines of u/v.

Euclidean metric (as often used in graphics), the weights w_{ij} are the well-known sum of the cotangents of the angles opposite to edge ij.

Vertices with Regular Continuity. When a vertex is on a *regular* singularity line between two patches, we assume that the fields u and v are smooth across the patch boundary *modulo a constant offset*. That is, if we call u^- (resp., v^-) the potential u of this vertex using its value from one of the patches, and u^+ (resp., v^+) the value at the same vertex but considering its value from the other patch, we wish to have:

$$u^- - u^+ = P_1 \qquad v^- - v^+ = P_2, \tag{1}$$

where P_1 and P_2 are two arbitrary integer constants associated with this particular patch boundary (we will discuss how to choose their values adequately later on). Obviously, enforcing this "equality modulo offset" will guarantee that integer isolines of u and v do match up at the boundary. Notice also that it corresponds to guaranteeing continuity of the 1-forms du and dv, as $d(u^+ - u^-) = du^+ - du^- = 0$. Finally, to ensure smoothness of these isolines, we enforce harmonicity of the two potentials taking the jump into account:

$$\sum_{j \in \mathcal{N}^-(i)} w_{ij} \begin{pmatrix} u_i^- - u_j \\ v_i^- - v_j \end{pmatrix} + \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} u_i^+ - u_j \\ v_i^+ - v_j \end{pmatrix} = 0$$

Fortunately, one realizes that the above conditions can be rewritten using only *one* value of u and *one* value of v for our boundary vertex i—therefore alleviating the need for storing two different values, one on each side of the singularity line. Indeed, if we assume $u_i \equiv u_i^-$, and thanks to Eq. (1):

$$\boxed{\sum_{j \in \mathcal{N}(i)} w_{ij} \begin{pmatrix} u_i - u_j \\ v_i - v_j \end{pmatrix} = \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}.}$$

Notice that this equation is a simple variant of the former case, modifying the right hand side to impose the correct conditions on each side of the boundary.

Vertices with Reverse Continuity. This time, we want the 0-forms u and v to change orientation when crossing the patch boundary. That is, we wish to have $du^+ = -du^-$, and $dv^+ = -dv^-$. These constraints are easily enforced by defining:

$$u^+ + u^- = Q_1 \qquad v^+ + v^- = Q_2,$$

where Q_1 and Q_2 are two integer constants associated to the boundary on which the vertex lies. We now enforce harmonicity of the two potentials at i *modulo* the reversal:

$$\sum_{j \in \mathcal{N}^-(i)} w_{ij} \begin{pmatrix} u_i^- - u_j \\ v_i^- - v_j \end{pmatrix} + \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} u_j - u_i^+ \\ v_j - v_i^+ \end{pmatrix} = 0$$

Again, one notices that a simpler expression using only one value for vertex i and a non-zero right-hand side, is:

$$\boxed{\sum_{j \in \mathcal{N}^-(i)} w_{ij} \begin{pmatrix} u_i - u_j \\ v_i - v_j \end{pmatrix} + \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} u_i + u_j \\ v_i + v_j \end{pmatrix} = \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}.}$$

This last expression preserves the *symmetric* nature of the Laplacian matrix. This is a particularly nice feature: state-of-the-art linear solvers have been shown to scale very well on such a problem.

Vertices with Switch Continuity. Finally, for vertices on a singularity line on which we want u and v to switch, we simply enforce that $du^+ = dv^-$ and $dv^+ = -du^-$. Notice the extra minus sign, because switching u and v reverses one of the two directions. Again, these conditions are satisfied if:

$$v^- - u^+ = R_1 \qquad v^+ + u^- = R_2,$$

Finally, to ensure smoothness of these isolines, we enforce harmonicity of both potentials given this discontinuity through:

$$\sum_{j \in \mathcal{N}^-(i)} w_{ij} \begin{pmatrix} u_i^- - u_j \\ v_i^- - v_j \end{pmatrix} + \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} v_j - v_i^+ \\ u_i^+ - u_j \end{pmatrix} = 0$$

The resulting symmetric expression, using only one value for the vertex i and a non-zero right-hand side, is now:

$$\boxed{\sum_{j \in \mathcal{N}^-(i)} w_{ij} \begin{pmatrix} u_i - u_j \\ v_i - v_j \end{pmatrix} + \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} u_i + v_j \\ v_i - u_j \end{pmatrix} = \sum_{j \in \mathcal{N}^+(i)} w_{ij} \begin{pmatrix} R_2 \\ R_1 \end{pmatrix}.}$$

Notice there is an analogous formula for what we could call reverse-switch continuity vertices, namely when we want to switch u and $-v$.

1.5 Properties of Singular Continuity

Although quite simple, the four cases we discussed above provide an already rich repertoire of singularities. In particular, the previously mentioned case of a genus-0 object with two poles can be handled quite simply by linking the two poles with a singularity line: this "virtual" cut on the sphere creates one single patch touching itself along a *regular continuity* boundary. Now, the two potentials u and v can be computed per vertex by solving a modified Laplace equation, with vertices along the singularity line having different coefficients and non-zero right-hand sides.

Independence of Boundary Positions. One remarkable property of the previous equations is that the exact position of the various boundaries between patches

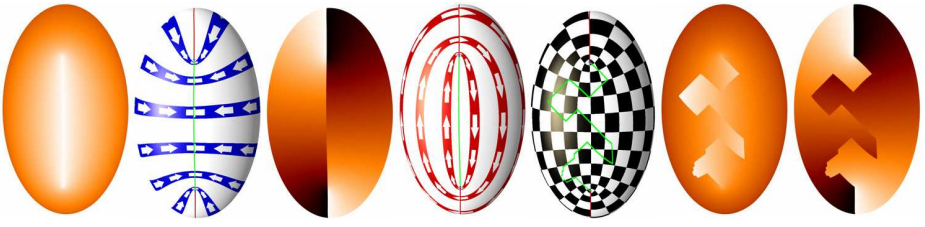


Fig. 4. Line Singularity. From left to right: Piecewise-continuous harmonic potentials u and v (color-shaded); Red and blue arrows depict the direction of the potential gradients; a checkerboard is mapped onto the ellipsoid using (u, v) as texture coordinates; when the singularity line is wiggly, the two potential functions change, but their isolines remain exactly identical to the previous case.

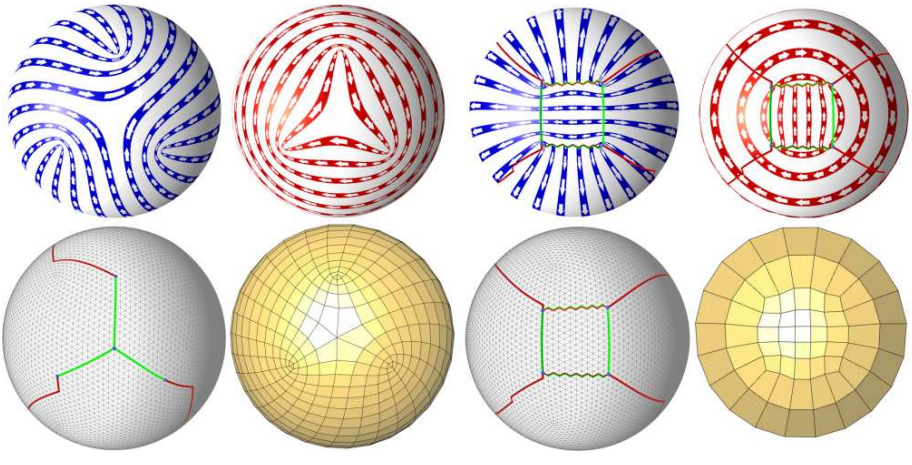


Fig. 5. Line Singularity as Basis of Many Singularities. Trisector (left) and square (right) singularities can be obtained by creating a graph of line singularities.

does *not* affect the final result: any boundary line in the same homology class as the original one will result in the same quad mesh. Although the 0-forms *will* be different (since their jumps will be located at distinct locations), their contouring will be exactly the same: only the local sign of their gradients will be affected in the reverse continuity case, while the gradient of u will become the gradient of v in the switch continuity case. In both cases, the union of the isolines of u and v remains the same! Therefore, the only real parameters are the set of

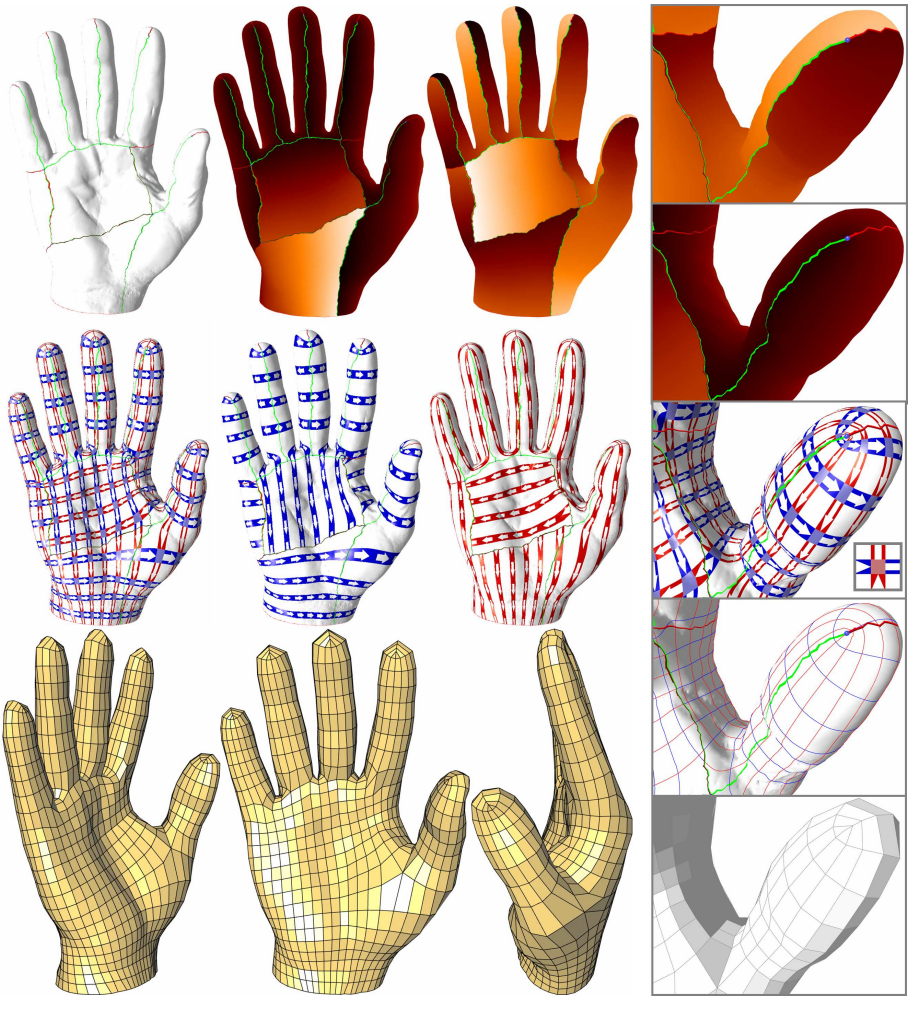


Fig. 6. Scanned Hand. From a triangulated surface and a set of line singularities assembled into a singularity graph, our technique solves a linear, modified Laplace equation to get two potentials (top); The pair of 1-forms associated to the potential differentials is specified as either regular, reverse or switch across *singularity lines* (center). An isocontouring of these potentials results in a pure-quad mesh with non-integer index singularities capturing the geometry (see close-up, right), and no T-junction (bottom).

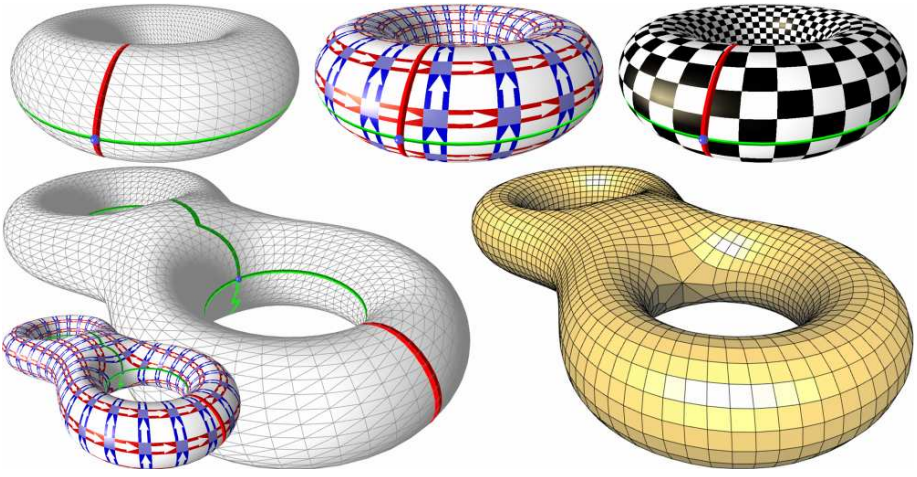


Fig. 7. Non-trivial Topology. Top: genus-1 example; Bottom: genus-2 model—a saddle (imposed by Hopf-Poincaré theorem) is present on a meta-vertex. The final meshes are still pure quad.

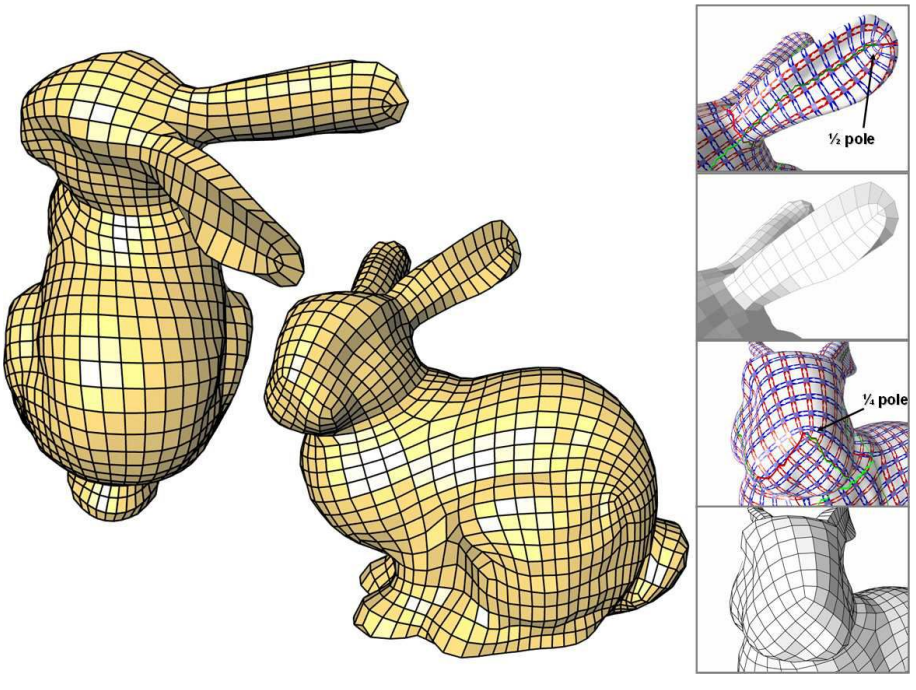


Fig. 8. Remeshing the Stanford Bunny. Left: Pure-quad remeshing. Right: detail of a half-pole, and a quarter-pole (using a switch and a regular line incident to the singularity). The half-pole becomes a degree-2 vertex, incident to two quads, with two nearly collinear edges. The degree-2 vertex can optionally be removed by merging its two incident quads into one.

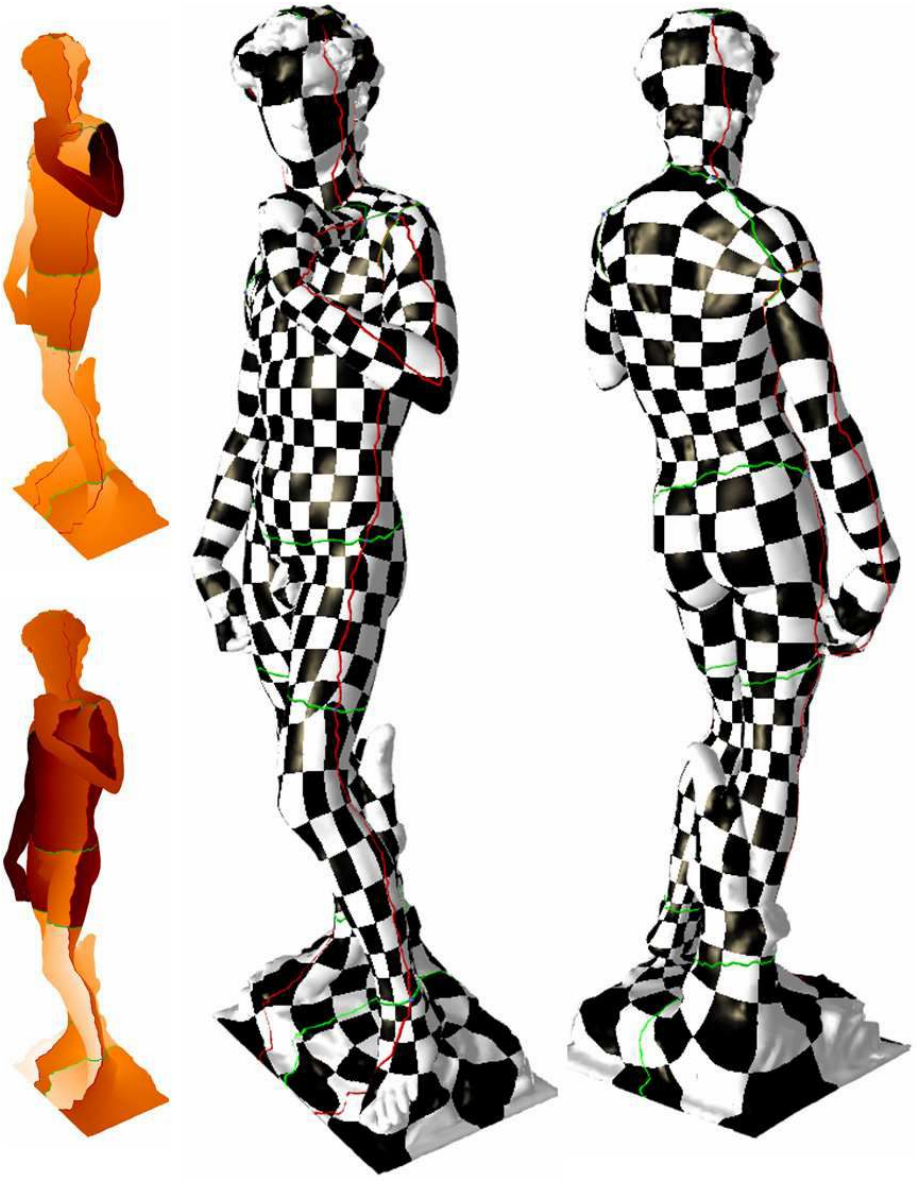


Fig. 9. Quadrangulating the Michelangelo David. The two potentials u and v are color-coded to reveal their piecewise continuous nature (left).

constants, chosen for each boundary (that we called P_1, P_2, Q_1, Q_2, R_1, and R_2 previously). This is quite convenient, as no special effort needs to be spent on getting "straight" boundary lines (see Figure 4, right). In other words, only the *topology* of the patches is needed.

Other Typical Singularities. Various other singularities can be achieved by designing a proper choice of boundary continuity between various patches. For instance, a trisector singularity, quite typical in direction fields, is obtained by assembling three concurrent lines, all of continuity type "reverse". A square singularity, i.e., four index-1/4 poles forming a square-shaped index-1 singularity, is assembled from four lines in the shape of a square, with type "regular", "switch", "reverse", and "switch" in cyclic order (see Figure 5). Notice that these cases create significantly less distortion, and by design, no T-junctions. We will provide, in the next section, a simple implementation method to handle all these singularities (and more) in a unified manner using a singularity graph; but the equations provided above can already accommodate all these cases.

2 Results

Figure 6 summarizes the main steps of our algorithm. Figure 7 shows meshes with non-trivial topology. Both singularity graphs were defined by homology generators. Figure 8 illustrates our algorithm on the Stanford Bunny model, and depicts details of half and quarter-poles. Figure 9 shows the quadrangulation of the Michelangelo David. The design of the David's singularity graph was done using a typical "clothing pattern" as used in garment design.

3 Conclusion

We have presented an approach for designing quadrangle tilings from arbitrary triangulated surface meshes. Our algorithm computes two piecewise smooth harmonic scalar functions, whose isolines tile the input surface into quadrangles, without any T-junctions. Our main contribution is an extension of the discrete Laplace operator which encompasses several types of line singularities. The resulting two discrete differential 1-forms are either regular, opposite or switched along the singularity graph edges. We show that this modification guarantees the continuity of the isolines across the lines, while the locations of the isolines themselves depend on the global solution to the modified Laplace equation over the whole surface. Design flexibility is provided through specification of the type of each line singularity of the graph, as well as the number of isolines along independent meta-edges to control quad sizes.

References

[ACSD+03] Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., Desbrun, M.: Anisotropic polygonal remeshing. ACM Trans. Graph. 22(3) (2003)

[AUGA05] Alliez, P., Ucelli, G., Gotsman, C., Attene, M.: Recent advances in remeshing of surfaces. STAR AIM@SHAPE (January 2005)

[BMRJ04] Boier-Martin, I., Rushmeier, H., Jin, J.: Parameterization of triangle meshes over quadrilateral domains. In: Symp. on Geometry processing, pp. 193–203 (2004)

[DBG+06] Dong, S., Bremer, P.-T., Garland, M., Pascucci, V., Hart, J.C.: Spectral surface quadrangulation. In: ACM SIGGRAPH '06, July 2006, ACM Press, New York (to appear)

[DKG05] Dong, S., Kircher, S., Garland, M.: Harmonic functions for quadrilateral remeshing of arbitrary manifolds. Computer Aided Design (Special Issue on Geometry Processing) 22(4), 392–423 (2005)

[Ede00] Edelsbrunner, H.: Mathematical problems in the reconstruction of shapes, Talk at MSRI's Workshop on Computational Algebraic Analysis (2000) http://msri.mathnet.or.kr/

[GH95] Grimm, C., Hughes, J.: Modeling surfaces of arbitrary topology. In: Proceedings of ACM SIGGRAPH, pp. 359–369 (July 1995)

[GY03] Gu, X., Yau, S.-T.: Global conformal parameterization. In: Symposium on Geometry Processing, pp. 127–137 (2003)

[JWYG04] Jin, M., Wang, Y., Yau, S.-T., Gu, X.: Optimal global conformal surface parameterization. In: IEEE Visualization, pp. 267–274. IEEE Computer Society Press, Los Alamitos (2004)

[KLS03] Khodakovsky, A., Litke, N., Schröder, P.: Globally smooth parameterizations with low distortion. ACM Trans. Graph. 22(3), 350–357 (2003)

[KSS06] Kharevych, L., Springborn, B., Schröder, P.: Discrete conformal mappings via circle patterns. ACM Trans. on Graphics 25(2) (2006)

[MK04] Marinov, M., Kobbelt, L.: Direct anisotropic quad-dominant remeshing. In: Proceedings of the Pacific Graphics, pp. 207–216 (2004)

[PP93] Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental Mathematics 2(1), 15–36 (1993)

[YZ04] Ying, L., Zorin, D.: A simple manifold-based construction of surfaces of arbitrary smoothness. ACM Trans. on Graphics 23(3), 271–275 (2004)

Surfaces with Piecewise Linear Support Functions over Spherical Triangulations

Henrik Almegaard[1], Anne Bagger[1], Jens Gravesen[2],
Bert Jüttler[3], and Zbynek Šír[4]

[1] Technical University of Denmark, Dept. of Civil Engineering/ [2] of Mathematics
[3] Johannes Kepler University, Institute of Applied Geometry, Linz, Austria
[4] Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic
aeb@byg.dtu.dk, hal@byg.dtu.dk, j.gravesen@mat.dtu.dk,
bert.juettler@jku.at, sir@karlin.mff.cuni.cz

Abstract. Given a smooth surface patch we construct an approximating piecewise linear structure. More precisely, we produce a mesh for which virtually all vertices have valency three. We present two methods for the construction of meshes whose facets are tangent to the original surface. These two methods can deal with elliptic and hyperbolic surfaces, respectively. In order to describe and to derive the construction, which is based on a projective duality, we use the so–called support function representation of the surface and of the mesh, where the latter one has a piecewise linear support function.

1 Introduction

Surface patches whose Gauss image (i.e., the mapping induced by the unit normals) defines a bijection between the surface and the co–domain of the Gauss image, which is a certain subset of the unit sphere, can be represented by their support functions. In this representation, a surface is described by the distance of its tangent planes to the origin of the coordinate system, which defines a function on the unit sphere. This representation is a fundamental tool in the field of convex geometry, see e.g. [2,5,6]. It has many potential applications in Computer Aided Design, as pointed out by Sabin [14].

Particular classes of support functions correspond to special types of surfaces. The case of surfaces with polynomial support functions has been discussed in [15]. As shown there, these surfaces are obtained by forming the convolution of certain surfaces of revolution whose meridians are special trochoids.

After polynomials, the simplest possible class of support functions are piecewise linear functions, which will be studied in this paper. These functions correspond to piecewise linear surfaces (meshes) with the property that the majority of vertices have valence three, and the majority of the facets are hexagons.

In order to solve the problem of reconstructing a mesh from its planar projections, similar meshes were constructed in [13] with the help of three–dimensional Voronoi diagrams. An optimization–based framework for applications in architecture was presented in [4]. By applying a projective duality to a mesh produced

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 42–63, 2007.
© Springer-Verlag Berlin Heidelberg 2007

by a subdivision surface, such meshes were produced in [9]. The relation between general polyhedra and their dual objects with respect to a projective duality (such as the polarity with respect to the unit sphere, which is sometimes called the "polar reciprocation") has been studied in classical geometry [3,16,17]. In particular, polyhedra with special properties, such as uniform polyhedra (where all stars of vertices have the same shape), have been analyzed. It should not be confused with the graph–theoretical duality, which is considered, e.g., in [11].

In this paper we use the support function to describe both the meshes and their dual objects. The support function is somehow "in–between" both objects, and it allows to describe both of them in a simple way. The use of the support functions automatically leads to a restriction of the class of dual meshes; these meshes have to be star–shaped with respect to the origin.

A possible application of the meshes described by piecewise linear support function comes from architecture, where free–form surfaces are often approximated by planar facets [4].

Motivated by applications in architectural design, constructions of quadrangular meshes with planar facets have recently been described in [10]. These meshes can be seen as discretizations of conjugate networks of curves on surfaces. In particular, as a discretization of the network of principal curvature lines, [10] introduces the class of conical (quadrangular) meshes. These meshes are characterized by the property that – for any offset distance – the four planes obtained by offsetting the planes spanned by the four quadrangles sharing a given vertex intersect in a single point. This is a desirable property for architectural design, since it facilitates the construction of offset (or parallel) structures, which may be needed for statical reasons. These meshes have further been analyzed in [12].

In the present paper we focus on meshes with planar faces, where virtually all vertices possess valency 3 (i.e., trihedral meshes). Consequently, most faces are planar hexagons. Clearly, any three planes parallel to the three faces (which are assumed to be mutually different) sharing a given vertex intersect in a single point, and the existence of offset structures is therefore guaranteed. The constructions described below are capable of producing meshes with exactly planar faces, as they work without numerical optimization.

The remainder of this paper is organized as follows. The next section presents some background information about shell structures in architectural design. In particular it discusses the different possible structures with planar facets. Section 3 introduces support functions and discusses their use for describing the polarity with respect to the unit sphere. The special case of piecewise linear support functions is described in Section 4. In particular, it is shown that these surfaces define a star–shaped triangular mesh and a primal mesh with quasi–convex facets, which are not always simple. Finally, in order to approximate a given surface by a mesh corresponding to a piecewise linear support function, the construction of tangent meshes is described in Section 5. In the case of hyperbolic surface patches, this can be achieved with the help of the parameterization by asymptotic lines, while elliptic surface patches can be dealt with via convex hull computations. The methods can be applied to surface patches whose Gauss

image defines a bijection between the patch and the corresponding spherical domain. In the case of elliptic surface patches, the spherical domain is required to be convex. Finally we conclude the paper.

2 Shell Structures for Architectural Design

Shell structures, and in particular their realizations as piecewise linear structures, are attractive tools for architectural design. We summarize the background from statics and discuss the case of faceted glass shells.

2.1 Shell Structures

Shell structures can in general be considered as structurally efficient structures. The efficiency is due to the fact that the curved form reduces the bending strength needed to carry the loads to almost zero, hence the shell surface can be very thin. For instance, an egg shell can take up considerable load as long as it is unbroken. It can drop from the nest without breaking. Only a concentrated point load – and especially a point load from the inside – will break it. Once it is broken, the egg shell is very weak. This is because the support conditions have changed, so that now only the bending strength is carrying the load.

In principle the bending strength can be zero, if the support conditions are appropriate. Then a specific ideal structural model, the membrane shell model, can be used. In the membrane shell model only membrane forces are considered. Membrane forces are in-plane shear and normal forces, and for a curved surface the considered plane is the tangent plane. Membrane forces are the type of forces that develop in tent structures, but in tents only tension forces can be developed. In membrane shell structures both tension and compression forces are developed. This means in principle that the shell surface should be considered perfectly bendable, both geometrically and statically. In order to keep the surface geometrically and statically determinate, the support conditions along the edge have to fulfill certain rules. Furthermore, the shell design has to follow certain guidelines in order to be stiff and to avoid too large internal stresses [1]. In practice though, shells have to have some bending stiffness in order to carry concentrated loads and to avoid instability from buckling.

Membrane shell structures can be designed either with smooth curved surfaces or with surfaces composed of plane facets - faceted surfaces (Fig. 1). From a construction point of view, the faceted surfaces are very attractive compared to the curved surfaces, as they are less complicated to build. The planar facets make them much easier to describe and produce industrially than doubly curved surfaces. At the same time, various standard materials and components can be used. For faceted surfaces, three geometrically and statically different systems are of interest.

– *Triangular facets* with six-way vertices. Statically this system generates concentrated forces along the edges, leading to the well known triangulated truss shell structures consisting of bars and joints (Fig. 2, left).

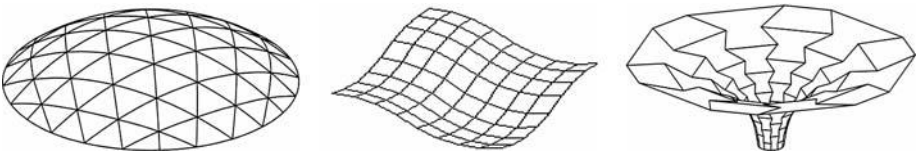


Fig. 1. Faceted shell structures. A) Triangulated truss shell structure. B) Quadrangular hybrid shell structure. C) Three-way vertices plate shell structure.

Fig. 2. Left: Triangulated *truss shell structure* consisting of bars and joints (Great Court, British Museum, London). Right: Hybrid type of structure consisting of a quadrangular truss stabilized by pre-stressed diagonal cables (Hippo House, Berlin Zoo).

- *Quadrangular facets* with four-way vertices. A widely used example is faceted translation shells (Fig. 2, right). Statically this system generates concentrated forces along the edges and shear forces in the facets, leading to a hybrid type of structure consisting of a quadrangular truss stabilized by plates or pre-stressed diagonal cables.
- *Hexagonal facets* with three-way vertices. Geometrically this system is dual to the triangulated system. Statically this system only generates in-plane shear and normal plate forces in the facets, leading to plate shell structures (Fig. 3, left).

The structural systems and forces here mentioned are the global structural systems and forces. They are all in-plane forces and equal to the membrane forces in smooth curved shells. This means that if the support conditions are appropriate, no bending stiffness is needed in the connections between the elements. Only locally the out of plane loads applied have to be carried to the edges of the facets by bending forces.

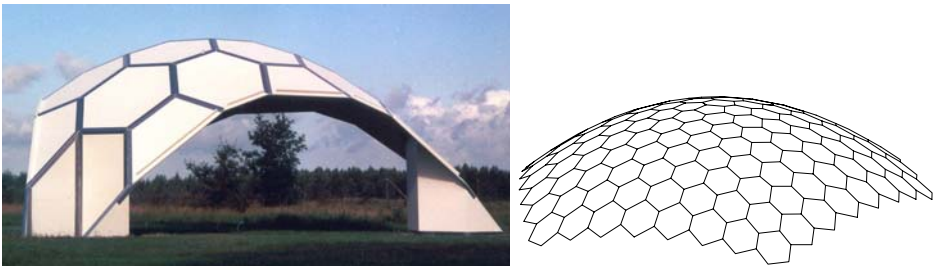


Fig. 3. Left: *Plate shell structure* consisting of plates primarily subjected to in-plane shear and normal forces (Full scale model, SBI Hørsholm). Right: Principal layout for a load carrying faceted shell structure of glass.

2.2 Faceted Glass Shell

Glass is already to some extent used for load carrying structural members like beams and columns. The structural use of glass is troubled by a brittle behavior of the material, and a limited capacity for carrying tension forces. However, these characteristics can be taken into account in the design process in various ways, thereby opening up for a world of transparent load carrying structures.

The load bearing capacity of glass is similar to that of wood when in tension, and to steel when in compression. In principle, glass is an exceptionally strong material, but in reality small flaws are distributed randomly over the surface. When the allowable tension stress is reached at the glass surface, one of these small flaws (a small crack) eventually will start to open. Since a redistribution of the stresses is not possible in glass, fracture will occur – hence the brittle behavior.

As described earlier, bending stresses are minimized in shell structures, which are appropriately shaped and supported, and the load is transferred primarily via in-plane stresses (membrane stresses). This allows for a better utilization of the capacity of the structural material, since stresses are distributed evenly over the thickness of the structure instead of concentrated at the surfaces. The stiffness to weight ratio of a shell structure – smooth or faceted – is remarkably good, since the absorption of loads is provided by the overall global shape of the structure, and not by a local sectional area. If structural instability can be avoided, the structural thickness of a smooth shell can easily be as little as 1/1000 of the span, or less. This is where glass becomes an interesting possibility as the load carrying material. A span of 15 meters corresponds to a glass thickness of 15 mm, if the thickness/span ratio of 1/1000 is achieved, and that is not an unrealistically large thickness for a laminated glass pane.

In order to avoid the high cost of the production of glass of double curvature, faceted glass shell structures are considered as an alternative to the smooth glass shell. Planar pieces of glass are more easily described, produced and transported. The glass thickness will increase somewhat compared to the smooth shell, since the structure will be burdened by unfavorable local effects.

A part of a faceted paraboloid of revolution is shown in Fig. 3, right. This geometrical drawing is in the following considered as the principle layout for a load carrying faceted shell structure of glass. The span is about 16 meters, and the size of the facets is roughly 1.2 meters. The glass elements are all planar, and the majority are shaped as hexagons. The central piece of glass is a pentagon, allowing the hexagons to keep roughly the same size, even though the structure is curved. The joints connecting the glass panes are designed to be able to transfer in-plane stresses.

3 Support Function and Dual Surface

We use the polarity with respect to the unit sphere to define the dual surface of a given surface and discuss its relation to the support function representation.

3.1 The Polarity with Respect to the Unit Sphere

Any non–degenerate quadric surface in three–dimensional space defines an associated *polarity*. In particular, we consider the unit sphere $\mathbb{S}^2$ (centered at the origin). The associated polarity π maps the point

$$\mathbf{p} = (p_1, p_2, p_3)^\top \tag{1}$$

into the plane

$$\mathbf{P} : \{\mathbf{x} = (x_1, x_2, x_3)^\top : \ 1 = \mathbf{p} \cdot \mathbf{x} = p_1 x_1 + p_2 x_2 + p_3 x_3\} \tag{2}$$

and vice versa, i.e., $\mathbf{P} = \pi(\mathbf{p})$ and $\mathbf{p} = \pi(\mathbf{P})$. It is defined for all points except for the origin, $\mathbf{p} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$, and for all planes which do not pass through the origin[1].

The polarity π preserves the incidence of points and planes: if a point $\mathbf{p}$ belongs to a plane $\mathbf{Q}$, then the plane $\pi(\mathbf{p})$ passes through the the point $\pi(\mathbf{Q})$.

The plane $\pi(\mathbf{p})$ is perpendicular to the line $\{\lambda\mathbf{p} : \lambda \in \mathbb{R}\}$. The distances d and D of the point and of its image plane from the origin satisfy $d\,D = 1$. In particular, each point of the unit sphere is mapped to the tangent plane at itself.

3.2 The Dual Surface

We consider a segment of a smooth ($\mathcal{C}^2$) surface $\mathbf{p}(u, v)$, $(u, v) \in \Omega$. Each point has an associated unit normal

$$\mathbf{N} : \ \Omega \to \mathbb{S}^2 : \ (u, v) \mapsto \mathbf{N}(u, v) \tag{3}$$

[1] These restrictions can be avoided by considering the projective closure of the three–dimensional space. The origin corresponds to the plane at infinity, and the infinite points correspond to planes passing through the origin.

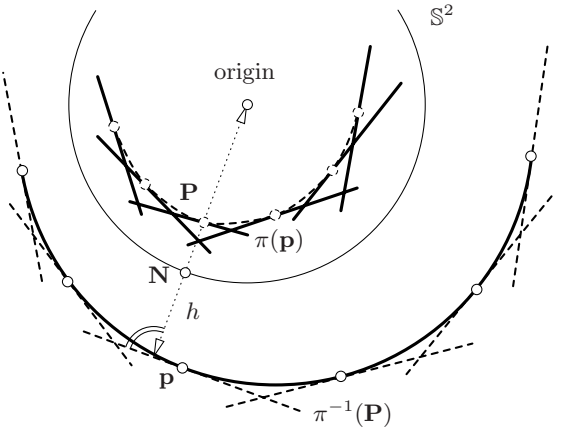


Fig. 4. The polarity π maps the points $\mathbf{p}$ of the primal surface to the tangents $\pi(\mathbf{p})$ of the dual surface (both shown as solid lines), and the tangent planes $\pi^{-1}(\mathbf{P}) = \pi(\mathbf{P})$ of the primal surface to the points $\mathbf{P}$ of the dual surface (both shown as dashed lines). The figure shows the two–dimensional situation.

which depends smoothly on u, v and defines an orientation of the surface. We assume that the mapping $\mathbf{N}$ is a bijective mapping between Ω and $\mathbf{N}(\Omega) \subseteq \mathbb{S}^2$. This is satisfied, provided that the surface does not contain parabolic or singular points and the domain is sufficiently small.

The polarity π is now applied to the *points* of the surface patch $\mathbf{p}(u, v)$. This produces the two–parameter family of planes

$$\{\mathbf{x} : \ 1 = \mathbf{x} \cdot \mathbf{p}(u, v)\}. \tag{4}$$

At the same time, the polarity is applied to the two–parameter family of *tangent planes* of the surface patch $\mathbf{p}(u, v)$. This leads to the points

$$\mathbf{P}(u, v) = \frac{1}{H(u, v)} \mathbf{N}(u, v), \tag{5}$$

where

$$H : \ \Omega \to \mathbb{R} : \ (u, v) \mapsto \mathbf{N}(u, v) \cdot \mathbf{p}(u, v) \tag{6}$$

measures the distance between the tangent plane and the origin.

These points form the *dual surface* of $\mathbf{p}(u, v)$ with respect to the polarity π, cf. Fig. 4. Its tangent planes are the images (4) of the points. The dual surface is well–defined provided that $H \neq 0$, i.e., no tangent plane of the surface patch $\mathbf{p}(u, v)$ contains the origin $\mathbf{0}$.

The signs of the Gaussian curvatures of the surface patch $\mathbf{p}(u, v)$ and of its dual surface $\mathbf{P}(u, v)$ are identical. If both surfaces have negative Gaussian curvature, then the asymptotic lines of $\mathbf{p}(u, v)$ correspond to the asymptotic lines of the dual surface.

Remark 1. Dual curves and surfaces have been considered by Hoschek [8] for detecting sign changes of the curvature and the Gaussian curvature. Hoschek uses the polarity with respect to the imaginary unit sphere, which corresponds to changing the sign of the right–hand side in (5).

3.3 Support Functions of Surfaces

The function

$$h : \mathbf{N}(\Omega) \to \mathbb{R} : \; h = H \circ \mathbf{N}^{-1} \tag{7}$$

which is obtained by composing the inverse of the map $\mathbf{N}$ defined in (3) with the function H defined in (6), is called the *support function* of the given surface patch. Support functions are a classical tool in the field of convex geometry [2,5,6]. Recently, curves and surfaces with polynomial support functions have been studied by three of the authors [15].

If a support function $h \in \mathcal{C}^1(D, \mathbb{R})$ is given, where $D \subseteq \mathbb{S}^2$, then the associated surface patch can be constructed by computing the envelope of the planes

$$\{\mathbf{x} : \; h(\mathbf{n}) = \mathbf{n} \cdot \mathbf{x}\}, \quad \mathbf{n} \in D. \tag{8}$$

For any $\mathbf{n} \in D$, the corresponding point on the envelope is

$$\mathbf{p}(\mathbf{n}) = h(\mathbf{n})\, \mathbf{n} + (\nabla_{\mathbb{S}^2} h)(\mathbf{n}) \tag{9}$$

where $\nabla_{\mathbb{S}^2}$ is the embedded intrinsic gradient of the support function h with respect to the unit sphere. If $h^* \in \mathcal{C}^1(\mathbb{R}^3, \mathbb{R})$ is a scalar field whose restriction to $\mathbb{S}^2$ equals h, then

$$(\nabla_{\mathbb{S}^2} h)(\mathbf{n}) = (\nabla h^*)(\mathbf{n}) - ((\nabla h^*)(\mathbf{n}) \cdot \mathbf{n})\, \mathbf{n}. \tag{10}$$

The mapping $\mathbf{n} \mapsto \mathbf{p}(\mathbf{n})$ is the inverse of the Gauss map of the surface patch.

The dual surface can also be obtained from the support function h,

$$\mathbf{P}(\mathbf{n}) = \frac{1}{h(\mathbf{n})} \mathbf{n}, \tag{11}$$

cf. (5).

Example 1. We consider the support function h which is obtained by restricting $h^* = 3 + 5n_1$ to the unit sphere $\mathbb{S}^2$. The intrinsic gradient (10) equals

$$(\nabla_{\mathbb{S}^2} h)(\mathbf{n}) = (5 - 5n_1^2, -5n_1 n_2, -5n_1 n_3)^\top. \tag{12}$$

Using (9) and (11) we obtain the equations of the primal and the dual surface,

$$\mathbf{p}(\mathbf{n}) = (3n_1 + 5, 3n_2, 3n_3)^\top, \quad \text{and} \quad \mathbf{P}(\mathbf{n}) = \frac{1}{3 + 5n_1} (n_1, n_2, n_3)^\top, \tag{13}$$

respectively. Now one may substitute any parameterization of the unit sphere for $\mathbf{n} = (n_1, n_2, n_3)^\top$ in order to parameterize these two surfaces. In this case – as for any linear polynomial h^* with non–vanishing constant term – we obtain a sphere and a quadric of revolution.

Remark 2. The surface defined by (9) is not always regular. If $h \in \mathcal{C}^2(D, \mathbb{R})$, then the regularity can be characterized by the intrinsic Hessian of h: If the mapping $(\mathrm{Hess}_{\mathbb{S}^2} + \mathrm{id})h$ has full rank, then the surface (9) is regular. See [15] for a detailed discussion of regularity.

Remark 3. The two support functions h and $-(\rho \circ h)$, where ρ is the reflection $\rho : \mathbf{n} \mapsto -\mathbf{n}$ with respect to the origin, define the same surface, but with different orientations.

4 Piecewise Linear Support Functions

We define piecewise linear functions on the unit sphere and discuss the primal and dual meshes associated with them.

4.1 Piecewise Linear Functions on Spherical Triangulations

Consider three linearly independent points $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{S}^2$ which are assumed to lie in one hemisphere of the unit sphere (i.e., there exists a vector $\mathbf{r} \in \mathbb{R}^3$ such that $\mathbf{r} \cdot \mathbf{v}_i > 0$, $i = 1, 2, 3$). The three great circular arcs connecting them, which are contained in the same hemisphere, bound a *spherical triangle*.

We consider a subset $D \subseteq \mathbb{S}^2$ which is bounded by a sequence of great circular arcs with vertices $\mathbf{n}_1, \ldots, \mathbf{n}_k$. In addition, let $\mathbf{n}_{k+1}, \ldots, \mathbf{n}_m \in \mathrm{int}D$ be additional points in the interior of D. A *spherical triangulation* $\mathcal{T}$ of D with vertices $(\mathbf{n}_i)_{i=1,\ldots,m}$ is a collection of spherical triangles satisfying the following three properties:

1. The interiors of any two spherical triangles are disjoint,
2. the intersection of any triangle with the set of vertices consists of exactly three points, which are the vertices of that triangle, and
3. the union of all triangles is equal to D.

Now we consider a spherical triangulation $\mathcal{T}$ with vertices $(\mathbf{n}_i)_{i=1,\ldots,m}$. We assume that each vertex $\mathbf{n}_i$ is equipped with an associated scalar value h_i. For any spherical triangle $\triangle_{ijk} \in \mathcal{T}$ with vertices $\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k$, we consider the unique homogeneous linear function

$$h^*_{ijk} : \mathbb{R}^3 \to \mathbb{R} : \mathbf{x} \mapsto (h_i, h_j, h_k)(\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k)^{-1}\mathbf{x}, \tag{14}$$

where $(\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k)$ is the 3×3–matrix with columns $\mathbf{n}_i, \mathbf{n}_j$ and $\mathbf{n}_k$, and restrict it to $\triangle_{ijk}$. This function satisfies $h^*_{ijk}(\mathbf{n}_l) = h_l$ for $l \in \{i, j, k\}$. The collection of all these functions defines a unique piecewise linear function $h \in \mathcal{C}(D, \mathbb{R})$ over the spherical triangulation which interpolates the given values,

$$h(\mathbf{n}_i) = h_i, \quad i = 1, \ldots, m. \tag{15}$$

In the remainder of this paper we assume that all h_i are positive,

$$h_i > 0, \quad i = 1, \ldots, m. \tag{16}$$

The piecewise linear interpolant h is then also positive for all $\mathbf{x} \in D$.

Remark 4. Index triples ijk as for $\triangle_{ijk}$ with the same set of indices, but different order, will be identified, i.e. $\triangle_{ijk} = \triangle_{ikj} = \triangle_{jik}$ etc. That is, ijk represents a subset of $\{1, \ldots, m\}$ with cardinality three, and these subsets are used to label the triangles, etc.

4.2 The Dual Mesh

With the help of the relationship between the dual surface and the support function, we now define the dual mesh via (11). More precisely, for each spherical triangle $\triangle_{ijk} \in \mathcal{T}$ with vertices $\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k$ we obtain a segment of the dual mesh,

$$\mathbf{P}_{ijk}(\mathbf{n}) = \frac{1}{h^*_{ijk}(\mathbf{n})}\mathbf{n}, \quad \mathbf{n} \in \triangle_{ijk}. \tag{17}$$

Using the parameterization

$$\mathbf{n}(u, v, w) = \frac{u\mathbf{n}_i + v\mathbf{n}_j + w\mathbf{n}_k}{||u\mathbf{n}_i + v\mathbf{n}_j + w\mathbf{n}_k||}, \quad u + v + w = 1;\ u, v, w \geq 0 \tag{18}$$

of the spherical triangle and by exploiting the fact that h^*_{ijk} is homogeneous, one gets the rational linear parameterization

$$\mathbf{P}_{ijk}(\mathbf{n}(u, v, w)) = \frac{u\mathbf{n}_i + v\mathbf{n}_j + w\mathbf{n}_k}{(h_i, h_j, h_k)(\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k)^{-1}(u\mathbf{n}_i + v\mathbf{n}_j + w\mathbf{n}_k)} \tag{19}$$

which describes the triangle with vertices

$$\mathbf{t}_l = \frac{1}{h_l}\mathbf{n}_l, \quad l \in \{i, j, k\}. \tag{20}$$

This leads to

Proposition 1. *The* ***dual mesh*** $\mathbf{P}$ *associated with the piecewise linear support function* $h \in \mathcal{C}(D, \mathbb{R})$ *satisfying* (15) *over the given spherical triangulation* $\mathcal{T}$ *of* D *with vertices* $(\mathbf{n}_i)_{i=1,\ldots,m}$ *is the triangular mesh with vertices* $(\mathbf{t}_i)_{i=1,\ldots,m}$*, which has the same connectivity as* $\mathcal{T}$*. This mesh is star–shaped with respect to the origin; each ray* $\lambda\mathbf{n}$ *with* $\lambda \geq 0$*,* $\mathbf{n} \in D$ *intersects the mesh* $\mathbf{P}$ *in a single point.*

4.3 Quasi–Convex Polygons

Before discussing the primal mesh, we consider planar curve–like objects with piecewise linear support functions.

Definition 1. *Consider a closed polygon with* v *vertices* $(\mathbf{q}_i)_{i=0,\ldots,v-1}$ *which lies in a plane* $\mathbf{T} \subset \mathbb{R}^3$*, where any triple of neighboring points is assumed to be non–collinear. For each edge*

$$E_i = \{(1 - t)\,\mathbf{q}_i + t\,\mathbf{q}_{i+1 \bmod v} :\ t \in [0, 1]\}, \quad i = 0, \ldots, v - 1, \tag{21}$$

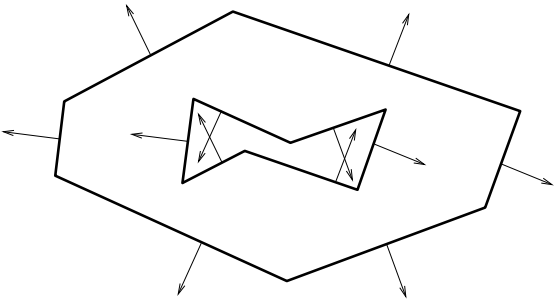


Fig. 5. Quasi–convex polygons as offsets of convex polygons

we choose one of the two possible unit normal vectors $\mathbf{n}_i$ *lying in* $\mathbf{T}$*. The polygon along with the unit normals is then called an* ***oriented polygon****. At each vertex* $\mathbf{q}_i$ *we consider the convex cone*

$$C_i = \{\lambda \mathbf{n}_{i-1} + \mu \mathbf{n}_{i \bmod v} : \ \lambda \geq 0, \mu > 0\}, \quad i = 1, \ldots, v. \tag{22}$$

The oriented polygon is said to be ***quasi–convex*** *if* $i \neq j$ *implies* $C_i \cap C_j = \emptyset$*.*

According to this definition, a quasi–convex polygon is characterized by the fact that the oriented unit normals trace the unit circle exactly once. Any convex polygon, where all edges are oriented by choosing either outward or inward pointing normals, is also quasi–convex. A general quasi–convex polygon is obtained as an offset of a convex one, where all edges are simply translated by a certain signed distance, according to the given normals.

Remark 5. If $\mathbf{T}$ is the plane $\mathbb{R}^2$, then an oriented polygon can be described by a piecewise linear support function on the unit circle $\mathcal{S}^1$. The edges and vertices of the polygon correspond to the nodes and to the linear pieces of this function, respectively.

4.4 The Primal Mesh

Each triangle $\triangle_{ijk} \in \mathcal{T}$ has an associated linear support function h^*_{ijk} which defines a triangular facet $\mathbf{P}_{ijk}$ of the dual mesh. By applying the polarity to the plane spanned by this facet, we obtain the corresponding point

$$\mathbf{p}_{ijk} = ((h_i, h_j, h_k)(\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k)^{-1})^\top \tag{23}$$

of the primal mesh.

For any inner vertex $\mathbf{n}_i \in \operatorname{int} D$ of the spherical triangulation we consider the *star* of this vertex, i.e., the set of triangles sharing this vertex,

$$\mathcal{S}_i = \{\triangle_{ijk} \in \mathcal{T} : \ j, k \in \{1, \ldots, m\}\}. \tag{24}$$

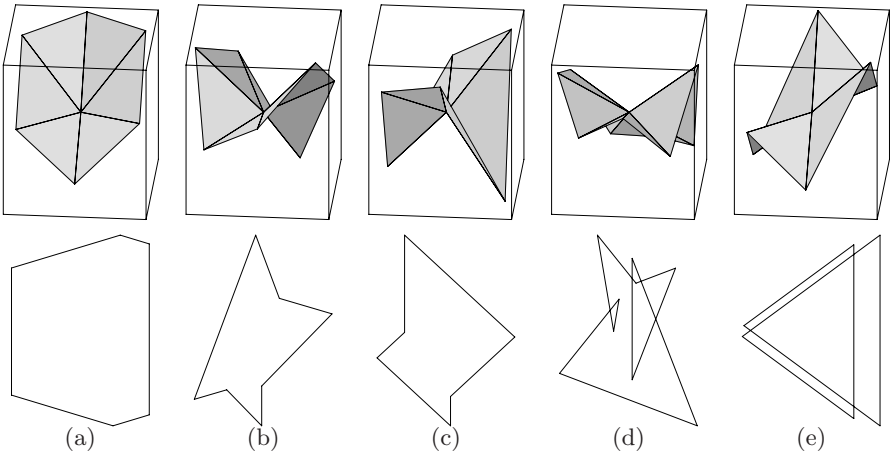


Fig. 6. Stars of vertices of the dual mesh and the associated quasi–convex polygons of the primal mesh: convex, simple (a); non–convex, simple (b,c); non–convex, non–simple (d,e)

Each triangle $\triangle_{ijk}$ corresponds to a point $\mathbf{p}_{ijk}$. All these points lie in the plane

$$\mathbf{T}_i = \{\mathbf{x} \in \mathbb{R}^3 : \ \mathbf{x} \cdot \mathbf{n}_i = h_i\}, \tag{25}$$

which is obtained by applying the polarity to the vertex $\mathbf{p}_i$. By connecting points, where the corresponding triangles have a common edge, we obtain a polygon.

Example 2. Five different stars and the associated planar polygons are shown in Fig. 6. If the vertex is convex, then the associated polygon is also convex. Otherwise, non–convex and even non–simple polygons may be obtained.

Proposition 2. *The* ***primal mesh*** $\mathbf{p}$ *associated with the piecewise linear support function* $h \in \mathcal{C}(D, \mathbb{R})$ *satisfying* (15) *over the given spherical triangulation* $\mathcal{T}$ *of* D *with vertices* $(\mathbf{n}_i)_{i=1,\ldots,m}$ *is the set of (not–necessarily simple) quasi–convex planar polygons with vertices* $\mathbf{p}_{ijk}$*. These polygons will be called the* ***facets*** *of* $\mathbf{p}$*. For each inner vertex* $\mathbf{n}_i$ *of the triangulation, the star of this vertex defines one of these polygons, and this polygon lies in the plane* $\mathbf{T}_i$*.*

Proof. Without loss of generality we consider the star of the vertex $\mathbf{n}_i = (0, 0, 1)^\top$. We consider a certain neighborhood of the corresponding vertex $\mathbf{t}_i = (1/h_i)\mathbf{n}_i$ of the dual mesh. Since it is star–shaped with respect to the origin, the dual mesh can be represented as

$$(1-t)\begin{pmatrix} 0 \\ 0 \\ 1/h_i \end{pmatrix} + t\begin{pmatrix} \cos\phi \\ \sin\phi \\ z(\phi) \end{pmatrix}, \quad \phi \in \mathbb{R}, t \in [0, \epsilon] \tag{26}$$

where the 2π–periodic continuous function $z(\phi)$ is piecewise smooth (in the interiors of the triangles $\mathbf{P}_{ijk}$). It has jumps in the first derivatives exactly in

those directions that correspond to edges shared by neighboring triangles of the dual mesh.

On the one hand, by applying the polarity to the planes spanned by the triangles $\mathbf{P}_{ijk}$ we obtain the vertices of the quasi–convex polygon in the plane $\mathbf{T}_i$. On the other hand, by applying the polarity to the parameter lines ϕ =constant in (26) we obtain oriented lines with the normal vector $(\cos\phi, \sin\phi, 0)^\top$ in this plane. In particular, the lines which correspond to shared edges of neighboring triangular facets of the dual mesh correspond to the edges of the quasi–convex polygon. The remaining lines support that polygon at its vertices. □

A simple polygon of the primal mesh will be said to be *regular*. A primal mesh with only simple polygons will also be said to be *regular*. We discuss criteria which guarantee regularity.

Lemma 1. *Let* $\mathbf{t}_i$ *be a an inner v-vertex of the dual mesh* $\mathbf{P}$ *and* $\mathbf{t}_{j(1)}, \ldots, \mathbf{t}_{j(v)}$ *be the neighboring vertices in counterclockwise order. We assume that no two edges meeting at* $\mathbf{t}_i$ *are linearly dependent.*

The quasi–convex polygon which corresponds to the star of $\mathbf{t}_i$ *has a self–intersection if and only if there exists two points* $\mathbf{t}_{j(k)}$, $\mathbf{t}_{j(l)}$ *such that the two points* $\mathbf{t}_{j(k-1)}, \mathbf{t}_{j(k+1)}$ *are on the same side of the plane spanned by* $\mathbf{t}_{j(k)}$, $\mathbf{t}_{j(l)}$ *and* $\mathbf{t}_i$*, and also the two points* $\mathbf{t}_{j(l-1)}, \mathbf{t}_{j(l+1)}$ *are on the same side of this plane.*

Proof. Applying the polarity to the plane spanned by $\mathbf{t}_{j(k)}$, $\mathbf{t}_{j(l)}$ and $\mathbf{t}_i$ gives the intersection point of the polygon. □

In particular, if the dual mesh is convex, then all facets of the primal mesh are also convex and therefore regular.

4.5 Examples

We use spherical triangulations whose vertices are obtained by applying uniform refinement steps (where each triangle is split into four triangles by halving the edges) to a icosahedron, and projection onto the unit sphere after each refinement step. This gives dual meshes which are adapted to the curvature of the mesh; more facets are used in regions with higher curvature.

Example 3. We consider an ellipsoid with three different diameters and its dual surface, which is again an ellipsoid. The vertices $\mathbf{n}_i$ of the spherical triangulation were obtained by applying two steps of refinement to a regular icosahedron. The three symmetry planes of the ellipsoid were aligned with symmetry planes of the icosahedron.

We considered only the upper half of the ellipsoid. The piecewise linear interpolant to the support function of the ellipsoid defines a dual mesh with 90 vertices. This dual mesh is convex, hence the 90 facets of the associated primal mesh do not have any self–intersections. A physical model of this mesh is shown in Figure 7.

Most facets of the primal mesh have 6 vertices, and most vertices of the primal mesh have valency 3. In some cases, the edges of the facets are very short, and

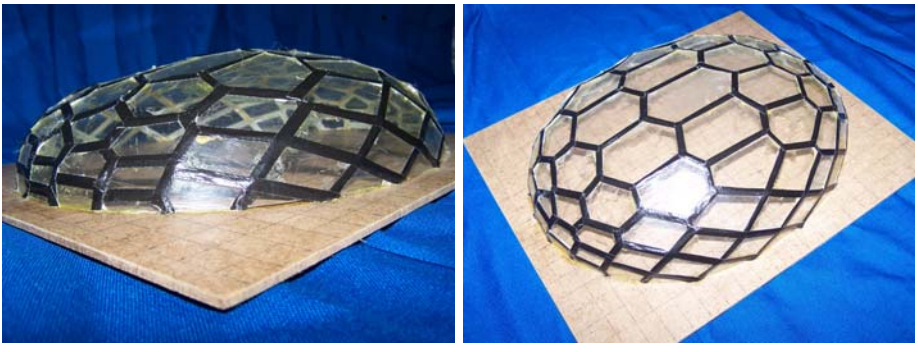


Fig. 7. Example 3. Convex mesh with piecewise linear support function which approximates a segment of an ellipsoid. The mesh consists of 90 facets.

two of these vertices are virtually identical, leading to a vertex with valency 4. This situation corresponds to two neighboring triangles of the dual mesh which are almost coplanar.

Example 4. We consider a one–sheeted hyperboloid with three different diameters and its dual surface, which is again a one–sheeted hyperboloid. Similarly to the previous example. the vertices $\mathbf{n}_i$ of the spherical triangulation were obtained by applying three steps of refinement (where each triangle is split into four triangles by halving the edges) to a regular icosahedron. The three symmetry planes of the hyperboloid were aligned with symmetry planes of the icosahedron.

We considered a segment of the upper half of the hyperboloid, which is bounded by two vertical planes. The piecewise linear interpolant to the support function of the hyperboloid defines a dual mesh with 169 vertices. This dual mesh is non–convex, but nevertheless each vertex defines a regular facet of the primal mesh. A physical model of this mesh is shown in Figure 8.

As in the previous example, most of the 169 facets of the primal mesh have 6 vertices, and most vertices of the primal mesh have valency 3.

5 Tangent Meshes

We discuss the construction of meshes whose facets lie in the tangent planes of a given surface. For a spherical triangulation with vertices $\mathbf{n}_1, \ldots, \mathbf{n}_k$ and a given (smooth) surface with support function $h(\mathbf{n})$, one can construct piecewise linear support function by choosing $h_i = h(\mathbf{n}_i)$ in (14), i.e., by sampling values of the support function of the given surface. However, we have to ensure the regularity of the corresponding primal mesh.

Two approaches will be presented. In the first approach, we consider a sequence of spherical triangulations with decreasing size of the triangles. They correspond to triangular (dual) meshes approximating the dual surface with increasing level of accuracy. Here we analyze the limit shapes of the facets of the

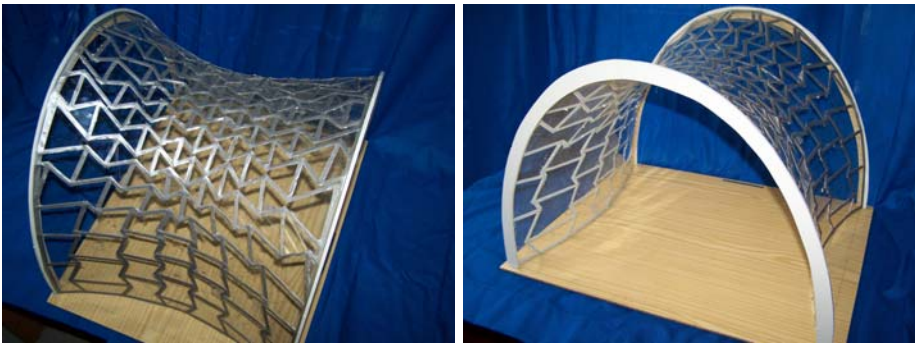


Fig. 8. Example 4. Non–convex mesh with piecewise linear support function which approximates a segment of a one–sheeted hyperboloid. The mesh has 169 facets.

primal mesh. This approach is particularly well suited for primal surfaces with only hyperbolic points, see Sections 5.1 and 5.2.

The second approach is restricted to primal surfaces with only elliptic points. In this case a suitable dual mesh can be found via convex hull computation, see Section 5.3.

5.1 Asymptotic Behavior of the Vertices of the Primal Mesh

We assume that we can generate an increasingly fine triangulation (the dual mesh) of the dual surface, which depends on some step-size h. In order to study the limit shape of the facets of the corresponding primal mesh when h tends to zero, we consider the following situation.

We consider the dual surface $\mathbf{P}$ in the vicinity of one of its points $\mathbf{P}_0 = \mathbf{P}(u_0, v_0)$. Consider two $\mathcal{C}^3$ curve segments $\mathbf{a}(h)$, $\mathbf{b}(h)$, $h \in (-\epsilon, \epsilon)$, which lie on $\mathbf{P}$ and satisfy $\mathbf{a}(0) = \mathbf{b}(0) = \mathbf{P}_0$ and $\mathbf{a}' \times \mathbf{b}' = \lambda \mathbf{n}_0$, $\lambda > 0$, where $\mathbf{n}_0$ is the normal of the dual surface at $\mathbf{P}_0$ and the prime $'$ denotes the first derivative with respect to h at $h = 0$. Let $\mathbf{Q}(h)$ be the plane spanned by $\mathbf{P}_0, \mathbf{a}(h), \mathbf{b}(h)$ and $\mathbf{q}(h)$ its image under the polarity with respect to the unit sphere tangent to $\mathbf{P}$ at $\mathbf{P}_0$ and having $\mathbf{n}_0$ for outer normal.

After a suitable translation, the sphere can be chosen as the unit sphere $\mathbb{S}^2$ centered at the origin, as described in Section 3.1. The plane $\mathbf{Q}(h)$ is well–defined for all $h \neq 0$, provided that ϵ is sufficiently small.

The limit behavior of $\mathbf{q}(h)$ for decreasing step-size h is described in the following result.

Lemma 2. *The point $\mathbf{q}(h)$ lies in the tangent plane to $\mathbf{P}$ at $\mathbf{P}_0$. It satisfies*

$$\lim_{h \to 0} \mathbf{q}(h) = \mathbf{P}_0 \quad \textit{and} \quad \mathbf{q}' = \frac{|\mathbf{b}'|^2 \kappa_{\mathbf{b}} \mathbf{a}' - |\mathbf{a}'|^2 \kappa_{\mathbf{a}} \mathbf{b}'}{2(\mathbf{a}' \times \mathbf{b}') \cdot \mathbf{n}_0} \times \mathbf{n}_0, \tag{27}$$

where $\kappa_{\mathbf{a}}$ and $\kappa_{\mathbf{b}}$ are normal curvatures of the tangent directions $\mathbf{a}'$ and $\mathbf{b}'$, respectively.

Proof. The two given curves have Taylor expansions of the form

$$\mathbf{a}(h) = \mathbf{P}_0 + \mathbf{a}'h + \frac{1}{2}(|\mathbf{a}'|^2 \kappa_{\mathbf{a}}\mathbf{n}_0 + \mathbf{t}_{\mathbf{a}})h^2 + \mathcal{O}(h^3) \tag{28}$$

$$\mathbf{b}(h) = \mathbf{P}_0 + \mathbf{b}'h + \frac{1}{2}(|\mathbf{b}'|^2 \kappa_{\mathbf{b}}\mathbf{n}_0 + \mathbf{t}_{\mathbf{b}})h^2 + \mathcal{O}(h^3), \tag{29}$$

where the vectors $\mathbf{t_a}$, $\mathbf{t_a}$ are perpendicular to $\mathbf{n}_0$. We compute a normal to the plane $\mathbf{Q}(h)$

$$\begin{aligned}\mathbf{N_Q}(h) &= (\mathbf{a}(h) - \mathbf{P}_0) \times (\mathbf{b}(h) - \mathbf{P}_0) = \\ &= (\mathbf{a}' \times \mathbf{b}') \left[h^2 + ch^3\right] + \tfrac{1}{2}(|\mathbf{b}'|^2 \kappa_{\mathbf{b}}\mathbf{a}' - |\mathbf{a}'|^2 \kappa_{\mathbf{a}}\mathbf{b}') \times \mathbf{n}_0 h^3 + \mathcal{O}(h^4),\end{aligned} \tag{30}$$

where

$$c = \frac{(\mathbf{a}' \times \mathbf{t}_b - \mathbf{b}' \times \mathbf{t}_a) \cdot (\mathbf{a}' \times \mathbf{b}')}{2|\mathbf{a}' \times \mathbf{b}'|^2}. \tag{31}$$

Since the plane $\mathbf{Q}(h)$ contains the point $\mathbf{P}_0 = \mathbf{n}_0$, it has the equation

$$\mathbf{x} \cdot \mathbf{N_Q}(h) = \mathbf{n}_0 \cdot \mathbf{N_Q}(h) = (\mathbf{a}' \times \mathbf{b}') \cdot \mathbf{n}_0 \left[h^2 + ch^3\right] + O(h^4). \tag{32}$$

Hence, the corresponding dual point is

$$\begin{aligned}\mathbf{q}(h) &= \frac{\mathbf{N_Q}(h)}{(\mathbf{a}' \times \mathbf{b}') \cdot \mathbf{n}_0 \left[h^2 + ch^3\right] + \mathcal{O}(h^4)} = \\ &= \mathbf{P}_0 + \frac{|\mathbf{b}'|^2 \kappa_{\mathbf{b}}\mathbf{a}' - |\mathbf{a}'|^2 \kappa_{\mathbf{a}}\mathbf{b}'}{2(\mathbf{a}' \times \mathbf{b}') \cdot \mathbf{n}_0} \times \mathbf{n}_0 h + \mathcal{O}(h^2).\end{aligned} \tag{33}$$

Though the calculation of $\mathbf{q}(h)$ is not valid for $h = 0$, the function $\mathbf{q}$ can be extended to a $\mathcal{C}^2$ function by letting $\mathbf{q}(0) = \mathbf{P}_0$. □

Using a regular parameterization $\mathbf{P} = \mathbf{P}(u, v)$ of the dual surface, we can express $\mathbf{q}'$ with the help of the second fundamental form.

Lemma 3. *Assume that* $\mathbf{P}_u(u_0, v_0) \times \mathbf{P}_v(u_0, v_0)$ *is a positive multiple of* $\mathbf{n}_0$, *where* $\mathbf{P}_u$, $\mathbf{P}_v$ *denote the first derivatives of* $\mathbf{P}$ *with respect to the parameters. If* $\mathbf{P}_0 = \mathbf{P}(u_0, v_0)$, $\mathbf{a}(h) = \mathbf{P}(u_a(h), v_a(h))$ *and* $\mathbf{b}(h) = \mathbf{P}(u_b(h), v_b(h))$ *then*

$$\begin{aligned}\mathbf{q}' = \mathbf{P}_u^{\perp} &\frac{L\left(u'_a {u'_b}^2 - {u'_a}^2 u'_b\right) + 2M u'_b u'_a \left(v'_b - v'_a\right) + N\left(u'_a {v'_b}^2 - {v'_a}^2 u'_b\right)}{2(u'_a v'_b - v'_a u'_b)} + \\ + \mathbf{P}_v^{\perp} &\frac{L\left(v'_a {u'_b}^2 - {u'_a}^2 v'_b\right) + 2M v'_a v'_b \left(u'_b - u'_a\right) + N\left({v'_b}^2 v'_a - v'_b {v'_a}^2\right)}{2(u'_a v'_b - v'_a u'_b)}\end{aligned} \tag{34}$$

where L, M, N *are the coefficients of the second fundamental form at* $\mathbf{P}_0$, *and*

$$\mathbf{P}_u^{\perp} = \frac{1}{\sqrt{EF - G^2}} \frac{\partial \mathbf{P}}{du}\bigg|_{u_0, v_0} \times \mathbf{n}_0, \quad \mathbf{P}_v^{\perp} = \frac{1}{\sqrt{EF - G^2}} \frac{\partial \mathbf{P}}{dv}\bigg|_{u_0, v_0} \times \mathbf{n}_0, \tag{35}$$

where E, F, G *are the coefficients of the first fundamental form at* $\mathbf{P}_0$.

This fact can now be verified by a direct computation.

5.2 Asymptotic Behavior of the Facets of the Dual Mesh

Eq. (34) gives the leading term of $\mathbf{q}(h)$ with respect to the basis $\mathbf{P}_u^{\perp}$, $\mathbf{P}_v^{\perp}$. We use it to define the limit shape of a facet.

Definition 2. *Consider the dual surface $\mathbf{P}(u,v)$, which is assumed to be regular and $\mathcal{C}^3$ in the vicinity of $\mathbf{P}(0,0)$. Let $\{(u_1,v_2),(u_2,v_2),\ldots,(u_n,v_n)\}$ be the parameter values of the star of the vertex $\mathbf{P}(0,0)$. We define the vertices of the* ***limit shape of the primal facet*** *by applying the expression* (34) *to the consecutive pairs of dual vertices. More precisely, the limit position of the vertex of the primal facet associated with the triangle with vertices*

$$\mathbf{P}(u_i,v_i), \quad \mathbf{P}(u_{(i \bmod n)+1}, v_{(i \bmod n)+1}), \quad \mathbf{P}(0,0), \quad i=1,\ldots,n, \tag{36}$$

is found by substituting

$$u'_a = u_i, \quad v'_a = v_i, \quad u'_b = u_{(i \bmod n)+1}, \quad \text{and} \quad v'_b = v_{(i \bmod n)+1} \tag{37}$$

in the right–hand side of (34)*, and the limit shape is obtained by connecting consecutive pairs of limit vertices.*

The geometrical meaning of the limit shape of the primal polygon is described in the following result.

Proposition 3. *We consider the primal facet associated with the star of the dual mesh with apex $\mathbf{P}(0,0)$ and vertices $\mathbf{P}(h\,u_i, h\,v_i)$, $i=1,\ldots,n$. As $h \to 0$, the shape of the primal facet tends to the corresponding limit shape.*

The proof is a direct consequence of Lemma 3.

By using a regular triangular mesh with edge–length h in the parameter domain of the dual surface, one might expect to obtain regular facets of the primal mesh as h tends to zero. However, this is not the case as shown by the following example.

Example 5. Consider the following elliptic dual surface

$$(u, v, 1.34u^2 + 1.89uv + 0.72v^2), \tag{38}$$

and choose the parameters (u_i, v_i) as the vertices of a regular hexagon in the u,v-plane inscribed in the unit circle. Figure 9 shows the regular polygon (thin) with its limit primal polygon (thick), which is not simple.

In order to obtain a regular primal mesh approximating a patch of of a hyperbolic surface, we propose the following

Algorithm 1
Input: Smooth dual hyperbolic surface $\mathbf{P}$ with a $\mathcal{C}^3$ curve $\alpha(t)$ lying on it, and step size h. The curve $\alpha(t)$ must not touch the asymptotic lines of the dual surface.
Output: Triangular (dual) mesh of the dual surface and associated primal mesh.

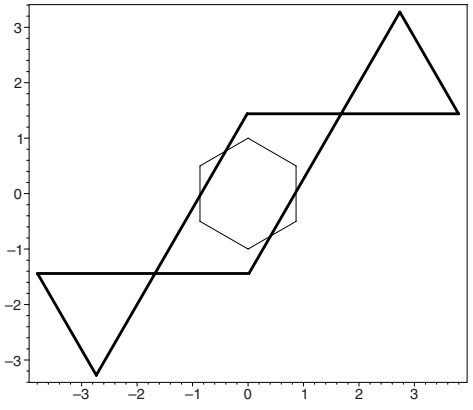


Fig. 9. The limit shape (thick polygon) associated with the star defined by the regular hexagon for the surface (38)

1. Compute the diagonal points $[n, n] := \alpha(nt)$.
2. Compute the general grid points $[m, n]$ defined as intersection of the 'vertical' asymptotic line through $[m, m]$ and the 'horizontal' asymptotic line through $[n, n]$. Here, the notions 'vertical' and 'horizontal' are used to distinguish between the two different families of asymptotic lines on the dual surface.
3. Produce the triangular mesh by applying the pattern shown in Figure 10, left.
4. Compute the primal mesh.

Now we can prove that this algorithm produces a sensible output, at least as h tends to zero.

Theorem 1. *If the step size h is sufficiently small, then the algorithm produces a regular primal mesh.*

Proof. There exists a unique parameterization $\mathbf{P}(u, v)$ of the dual surface such that the the parametric directions are the asymptotic lines and $\alpha(t) = \mathbf{P}(t, t)$. In this parameterization, the second fundamental form satisfies $L = N = 0$ and we can compute the limit shape of the primal facets associated with the stars of the dual mesh (indicated by the grey hexagons in Fig. 10, left). In this parameterization we apply directly the formula (34). Note that $L = N = 0$. Setting $M = 1$ we obtain the regular limit shape shown in Figure 10, right.

Different values of M only scale the limit shape. Note that this shape corresponds to the polarity with respect to a unit sphere which is tangent to the dual surface at the corresponding vertex. The global polarity π, however, modifies this shape by a projective transformation which can be shown to preserve the regularity, provided that the step–size is sufficiently small. □

We illustrate this result by two examples.

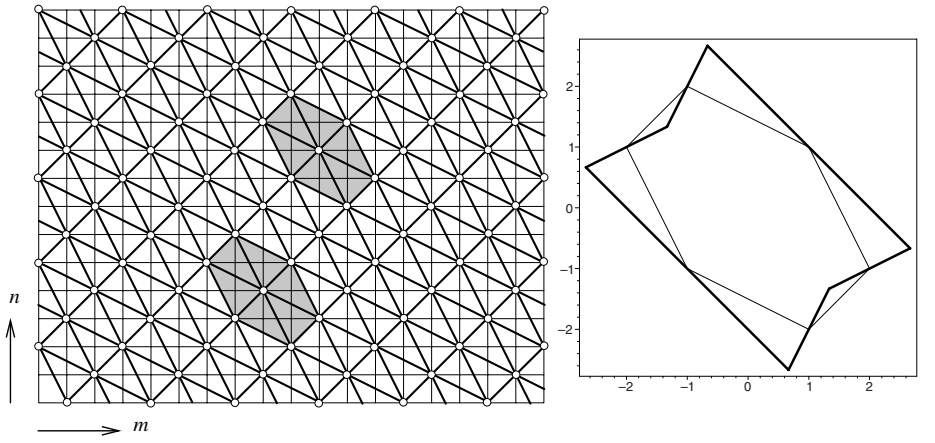


Fig. 10. Left: Pattern for producing the dual mesh from the dual surface. The vertical and horizontal lines represent the grid of asymptotic lines, while the circles indicate which points should be sampled. Right: The limit shape (thick curve) of the star defined by the thin polygon, which corresponds to the hexagons shown in the left figure.

Example 6. Consider the Enneper surface

$$\left(\tfrac{1}{3}(u+v)(2u^2-8uv+3+2v^2), \tfrac{1}{3}(u-v)(2u^2+8uv+3+2v^2), 4uv+1)\right),$$

where the parameter lines are already the asymptotic lines. The dual surface has the parametric representation

$$\mathbf{P}(u,v) = \begin{pmatrix} \frac{6(u+v)}{(8u^3v+6u^2+8uv^3+12uv+6v^2-3)} \\ \frac{-6(-v+u)}{(8u^3v+6u^2+8uv^3+12uv+6v^2-3)} \\ \frac{3(2v^2+2u^2-1)}{(8u^3v+6u^2+8uv^3+12uv+6v^2-3)} \end{pmatrix}. \tag{39}$$

By choosing $\alpha(t) = \mathbf{P}(t,t)$ and suitable h we obtain a dual mesh of $\mathbf{P}$ and finally a regular primal mesh which approximates the Enneper surface – see Figure 11.

This example is somewhat special, since the asymptotic lines intersect each other orthogonally (as it is a minimal surface). This is not the case of the second example.

Example 7. We consider a segment of a ruled quadric, where the parameter lines are the two families of straight lines on this surface. By applying the algorithm with three different step-sizes we obtain the primal meshes shown in Fig. 12.

Clearly, this technique can also be used to analyze other patterns than the one shown in Fig. 10, including non–uniform ones. This will be a topic of future research.

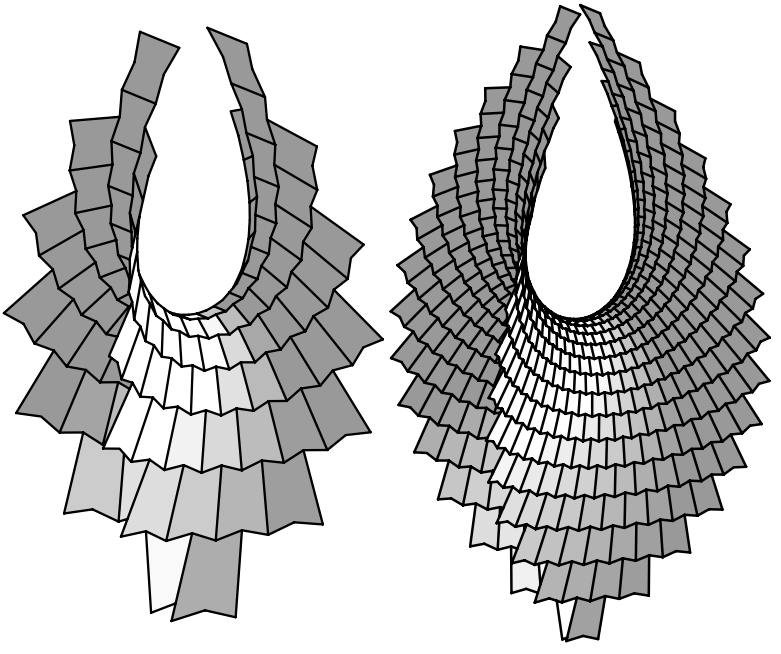


Fig. 11. Primal tangent meshes of the Enneper surface for values $h = 0.081$ (left) and $h = 0.039$ (right)

5.3 The Case of Elliptic Surfaces

If the given primal surface patch has only elliptic points, then a different algorithm can be used. We assume that the Gauss image of the given patch is contained in a hemisphere. In addition, it should be convex, i.e., any great circular arc connecting two points of the Gauss image should entirely belong to the Gauss image, too.

An approximation of the primal surface by a regular mesh can be obtained as follows.

Algorithm 2

1. Choose a spherical triangulation of the sphere.
2. For all vertices $\mathbf{n}_i$, $i = 1, \ldots, m$, evaluate the support function of the given surface, $h_i = h(\mathbf{n}_i)$ and compute the corresponding points $\mathbf{t}_i = \frac{1}{h_i}\mathbf{n}_i$ of the dual mesh.
3. Compute the convex hull of the points $\{\mathbf{t}_i, i = 1, \ldots, m\}$.
4. The apparent contour of the convex hull, seen from the origin, splits the boundary surface of the convex hull into two meshes. One of them is the suitable dual mesh; the other one does not have any inner vertices.

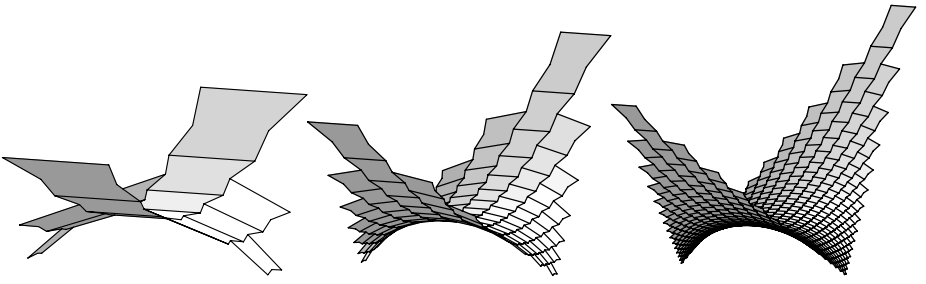


Fig. 12. Primal tangent meshes of a ruled quadric for three different values of h

Due to the fact that the dual surface of the elliptic point is also elliptic, all vertices of the dual mesh obtained from this algorithm have convex stars. Consequently, all faces of the primal mesh are convex (and therefore simple) polygons.

An example has already been presented in Section 4.5 (Example 3).

6 Conclusion

Based on the use of piecewise linear support functions we discussed dual meshes, which were assumed to be star–shaped with respect to the origin, and the associated primal meshes. The primal meshes are capable of approximating smooth surface patches without parabolic points. It should be noted that these meshes approximate not only the surfaces but also the associated normals. I.e., each normal in the Gauss image corresponds to exactly one normal on the primal mesh. This is clearly not the case for general meshes which approximate a given surface.

As a matter of future work we will use the support function for generating error bounds. In the smooth case the maximum distance error is essentially equal to the maximum difference between the original support function and its approximation. While this is also true for convex primal meshes, the extension of this result to the non–convex case is still an open problem.

In addition, we plan to discuss the approximation of general support functions by piecewise linear support functions over a given spherical triangulation, subject to conditions which guarantee the regularity of the resulting primal mesh. In the case of elliptic surfaces, this can be formulated as an optimization problem with linear constraints. In the hyperbolic case, however, non–linear constraints are needed.

Finally we plan to investigate surfaces with parabolic lines separating elliptic and hyperbolic regions. In order to represent these surfaces, multi–valued piecewise linear functions will be needed.

Acknowledgment. This research was supported by a grant of the Austrian Science Fund (FWF, project no. P17387-N12 and SFB F013, subproject 15), by

research project no. MSM 0021620839 at Charles University, Prague, and by the project "Facetted glass shells" at the Department of Civil Engineering of Danish Technical University. The authors wish to thank the reviewers for their useful comments. Special thanks go to Bert's father for building the two models shown in Figures 7 and 8.

References

1. Almegaard, H.: The Stringer System—a Truss Model of Membrane Shells for Analysis and Design of Boundary Conditions. Int. J. Space Structures 19, 1–10 (2004)
2. Bonnesen, T., Fenchel, W.: Theory of Convex Bodies. BCS Associates, Moscow, Idaho (1987)
3. Brückner, M.: Vielecke und Vielflache—Theorie und Geschichte. Teubner, Leipzig (1900)
4. Cutler, B., Whiting, E.: Constrained Planar Remeshing for Architecture. In: Symposium on Geometry Processing 2006, Poster proceedings (electronic), p. 5 (2006), http://sgp2006.sc.unica.it/program/PosterProceedings.pdf
5. Groemer, H.: Geometric Applications of Fourier Series and Spherical Harmonics. Cambridge University Press, Cambridge (1996)
6. Gruber, P.M., Wills, J.M. (eds.): Handbook of Convex Geometry. North-Holland, Amsterdam (1993)
7. Hoschek, J., Lasser, D.: Fundamentals of Computer Aided Geometric Design. AK Peters, Wellesley, Mass (1996)
8. Hoschek, J.: Dual Bézier Curves and Surfaces. In: Barnhill, R.E., Boehm, W. (eds.) Surfaces in Computer Aided Geometric Design, pp. 147–156. North-Holland, Amsterdam (1983)
9. Kawarahada, H., Sugihara, K.: Dual Subdivision: A New Class of Subdivision Schemes using Projective Duality. In: Jorge, J., Skala, V. (eds.) Proc. WSCG'2006, pp. 9–16. University of West Bohemia, Plzen (2006)
10. Liu, Y., Pottmann, H., Wallner, J., Yang, Y., Wang, W.: Geometric Modeling with Conical Meshes and Developable Surfaces. ACM Trans. Graphics 25, 681–689 (2006)
11. Patanè, G., Spagnuolo, M.: Triangle Mesh Duality: Reconstruction and Smoothing. In: Wilson, M.J., Martin, R.R. (eds.) Mathematics of Surfaces. LNCS, vol. 2768, pp. 111–128. Springer, Heidelberg (2003)
12. Pottmann, H., Wallner, J.: The Focal Geometry of Circular and Conical Meshes. Adv. Comput. Math. (to appear)
13. Ros, L., Sugihara, K., Thomas, F.: Towards Shape Representation using Trihedral Mesh Projections. The Visual Computer 19, 139–150 (2003)
14. Sabin, M.: A Class of Surfaces Closed under Five Important Geometric Operations. Technical Report VTO/MS/207, British Aircraft Corporation (1974), Available at http://www.damtp.cam.ac.uk/user/na/people/Malcolm/vtoms/vtos.html
15. Šír, Z., Gravesen, J., Jüttler, B.: Curves and surfaces represented by polynomial support functions. SFB report no. 2006-36 (2006), Available at http://www.sfb013.uni-linz.ac.at
16. Weisstein, E.W.: Dual Polyhedron. From MathWorld — A Wolfram Web Resource. http://mathworld.wolfram.com/DualPolyhedron.html
17. Wenninger, M.J.: Dual Models. Cambridge University Press, Cambridge (1983)

A Developable Surface of Uniformly Negative Internal Angle Deficit

Phillips A. Benton

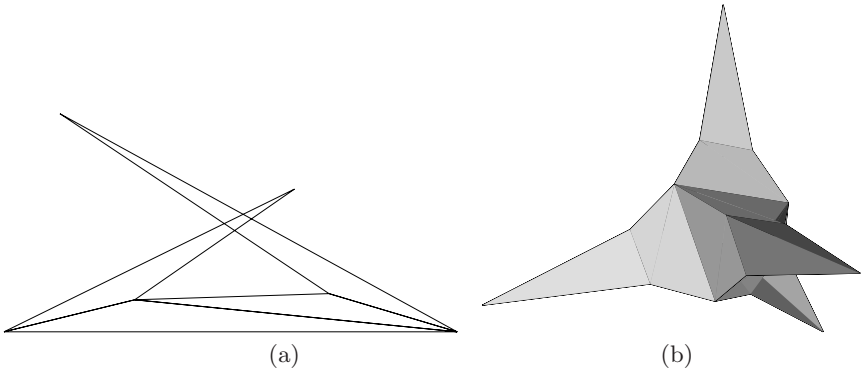
Centre for Mathematical Sciences, Cambridge CB3 0WA, United Kingdom
A.Benton@damtp.cam.ac.uk

Abstract. The author addresses the Edge Unfolding Problem, the task of determining whether a given polyhedral mesh may be cut at its edges and developed into a planar polyhedral net. It is shown that simply-connected surfaces of negative interior curvature cannot be developed. An example is then given of a surface of negative interior curvature with two boundary loops, isomorphic to a cylinder, which is developable.

1 Introduction

The *Unfolding Problem* was formally posed as an open question by Shepherd in [S75], but unfoldings of polyhedral surfaces can be found as far back in published history as 1525 in the work of the German artist/mathematician Albrecht Dürer [S77] (pp. 316-347, 414-417). While a number of advances have refined the more general problem (Bern et al in [BDEK99], Benbernou et al in [BCO04], etc) the fundamental question of edge-unfolding remains unsolved: how to determine the set of edges of a polyhedral model to cut, allowing the model to unfold into a flattened, non-self-intersecting form.

It is widely supposed that all convex polyhedra can be unfolded ([O98], p. 2). Small convex models have been identified which exhibit at least one illegal (self-intersecting) unfolding; Namiki and Fukuda [NF94] found the elegant example



(a) (b)

Fig. 1. (a) Namiki and Fukuda's overlapping tetrahedral unfolding; (b) an assembly of four instances of Bern et al's undevelopable Witch's Hat model

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 64–77, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of the slim tetrahedron (Figure 1a.) No example has been found yet of a convex surface which admits no legal unfolding. In contrast, non-convex surfaces which cannot be unfolded are well-known, such as Bern et al's 'witch's hat' model [BDEK99] (Figure 1b.)

2 Definitions

A polyhedral surface is said to be *developable* if a subset of its edges can be found which may be cut such that the faces of the mesh remain connected by a net of edges about which the mesh may be unfolded, flattening to the plane without overlap.

An edge or series of edges broken in the course of developing a model is referred to as a *cut*.

Cuts join together to form the *cut graph*, a connected undirected graph of broken edges. It will be shown that there are restrictions on the number of loops in the cut graph, and on the number of loops which may be formed by taking the union of the cut graph with the graph of the boundary edges of the source mesh. A *branch* in the cut graph is any node of valency greater than two.

The dual of the cut graph is the *unfolding tree*, the undirected, connected, acyclic and planar graph of the connectivity of the net of faces which remains after all cuts have been made. The unfolding tree is a subset of the graph of connectivity of faces of the original source mesh.

The Gaussian curvature of the surface of any polyhedral mesh is zero everywhere except at the vertices, where it is infinite. A number of approaches have been devised for determining the discrete curvature of a surface at a vertex, and Meyer et al provide a very accessible description of the Gauss-Bonnet scheme in [MDSB03]. A simplified form of the Gauss-Bonnet scheme is the *angle deficit* method, well-described by Van Loon in [V94] (p.5). The angle deficit of a vertex V is 2π minus the sum of the corner angles of the polygonal facets meeting at V.

3 Cycles in the Cut Graph, Angle Deficit and Developability

By definition, the unfolding tree cannot possess cycles and must be connected. The cut graph may possess cycles; the circumstances in which this is permitted may be bounded explicitly:

Lemma 1. *The presence of a loop in the cut graph requires that there be at least one handle in the topology of the source mesh.*

Proof: A loop in the cut-graph of the surface forms a Jordan curve on the surface which separates any locally $\Re_2$ surface into exactly two discrete components[1]. If

[1] From [G76], Appendix A.2, p.95. Griffiths does not prove this theorem directly, instead citing Newman's *Elements of the Topology of Plane Sets of Points*, Cambridge University Press, London (1954), p. 137.

the source mesh is without handle (locally $\Re_2$) then the unfolding tree will be broken into two parts. Therefore either the surface must have a handle which connects the faces within the loop to the faces outside the loop; or, the cut-graph cannot loop. □

Corollary 1. *The formation of a loop in taking the union of the cut graph with the set of all boundary edges of the source mesh requires that there be at least one handle in the topology of the source mesh.*

Proof: Boundary edges, like cut edges, have no dual in the unfolding tree. Loops in the (potentially disjoint) graph of boundary edges are permissible but, as in Lemma 1, new loops may not be introduced in taking the union of the cut graph and the boundary graph on a simply-connected surface. To do so would separate the surface into two or more disjoint parts as above. □

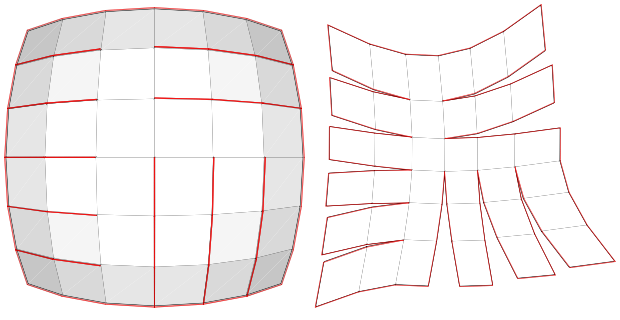


Fig. 2. Positive curvature: edge cuts terminate at inner vertices

There is a direct correspondence between the sign of the angle deficit at a vertex and the number of cuts it requires [P03]:

- At each spherical[2] vertex there must be at least one cut;
- At each hyperbolic[3] vertex there must be at least two cuts which take off one or more faces during the unfolding.

Vertices of zero angle deficit may be left completely untouched; they can flatten to the plane without cutting a single incident edge.

There can never be a leaf node (endpoint) of the cut graph on a vertex of negative angle deficit, because if at least two edges incident to a negative-curvature vertex must be cut then no such vertex can ever be a leaf node of the cut tree. If a branch of the cut-tree were to terminate at a negative-curvature vertex, only one incident edge would be broken.

[2] Angle deficit greater than zero.
[3] Angle deficit less than zero.

This may be summarized as

Lemma 2. *No simply-connected surface of uniformly negative internal angle deficit is developable.*

Proof: Suppose that there exists a simply-connected surface of uniformly negative internal angle deficit which is developable. To generate the unfolding, a cut graph must exist. This graph cannot loop, as the surface is simply connected; likewise, it may contain at most one vertex which lies on a boundary edge of the surface . If the graph does not loop then it must have at least two leaf nodes, where the edges of the cut graph originate and terminate, but all vertices have negative curvature. Therefore no such surface may exist. □

Corollary 2. *The number of branches in the cut-tree which develops a mesh of uniformly negative internal angle deficit cannot exceed $B - 2$, where B is the number of distinct boundary curves on the mesh.*

Proof: The number of leaf nodes of a tree is two plus the sum of the valence minus one of each branch node in the tree. Thus each branch raises the required number of boundary curves by one. As above, there must be at most one leaf node per boundary curve and no leaf node may not fall on a boundary curve. □

Corollary 3. *The (topologically) simplest developable surface of uniformly negative internal angle deficit has at least two boundary curves.*

Proof: It would be impossible for a graph to have -1 branch points. □

4 A Developable Surface of Uniformly Negative Internal Angle Deficit

A *coolinoid*[4] is the surface of rotation obtained by rotating a polynomial $f(t) = t^k + 1$ about the t axis. To obtain a discrete representation, the following parametric description is used:

```
 for (u:=0 to 1 step 1/du)
for (v:=-1 to 1 step 2/dv)
  x = cos(2πu) * (v^k + 1);
  y = v * h;
  z = sin(2πu) * (v^k + 1);
```

where $k \geq 1$ and $h > 0$. The model shown in Figure 3 was generated from ($h = 10.0$, $k = 2.2$, $du = 25$, $dv = 25$). Informally, h may be thought of as the 'stretch' of the model, k may be seen as the 'curviness' of the model and du and dv are the 'resolution' of the mesh.

[4] So named for its resemblance to the cooling tower of a power plant.

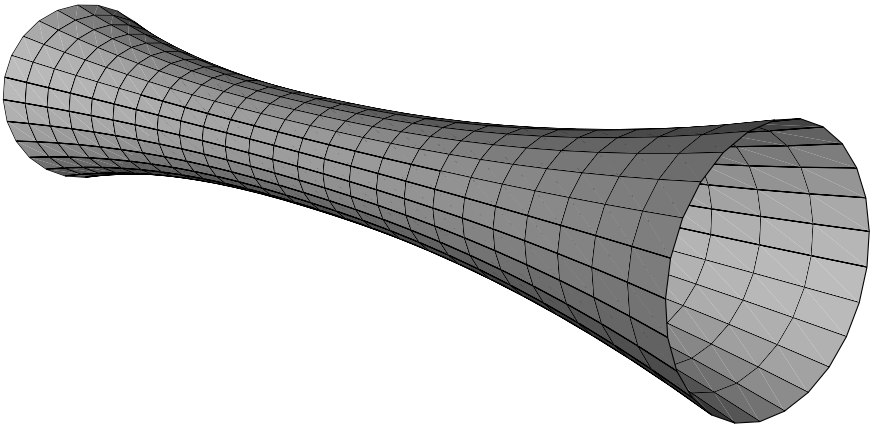


Fig. 3. Coolinoid

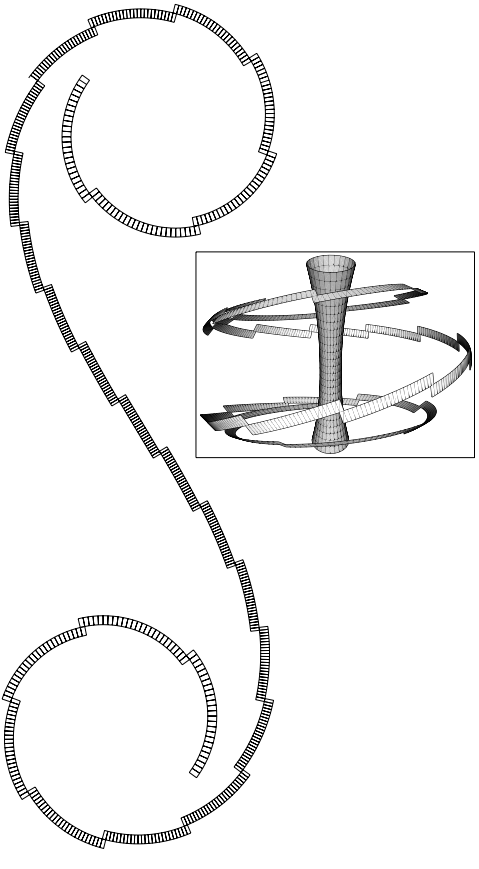


Fig. 4. Spiral unfolding of a coolinoid ($h = 10.0$, $k = 2.2$, $du = 25$, $dv = 25$.) Inset: The spiral unfolding in progress.

Many surfaces whose internal angle deficit is negative at every vertex are not developable, and several examples of such surfaces are shown in Figure 5. However, *there exist configurations of the coolinoid which are developable.* The model shown in Figure 3 unfolds into the spiraling net shown in Figure 4. The net has no self-intersection and is completely planar.

This demonstrates that

Theorem 1. *There exists a connected surface of uniformly negative internal angle deficit which is developable.*

Proof: By example (Figure 4.)
Note: Numerical verification that the model shown in Figure 4 has no self-intersection may be obtained by verifying that Equations 1 and 2, below, have no common solution for $h = 10.0$, $k = 2.2$, $du = 25$, $dv = 25$. □

5 A Black-Box Solution for Determining the Unfoldability of a Coolinoid

The observation that developability of the coolinoid varied as a function of h, k, du, dv spurred the development of a 'black-box' solution which could determine *a priori* whether a given coolinoid model could be unfolded. Note that for general

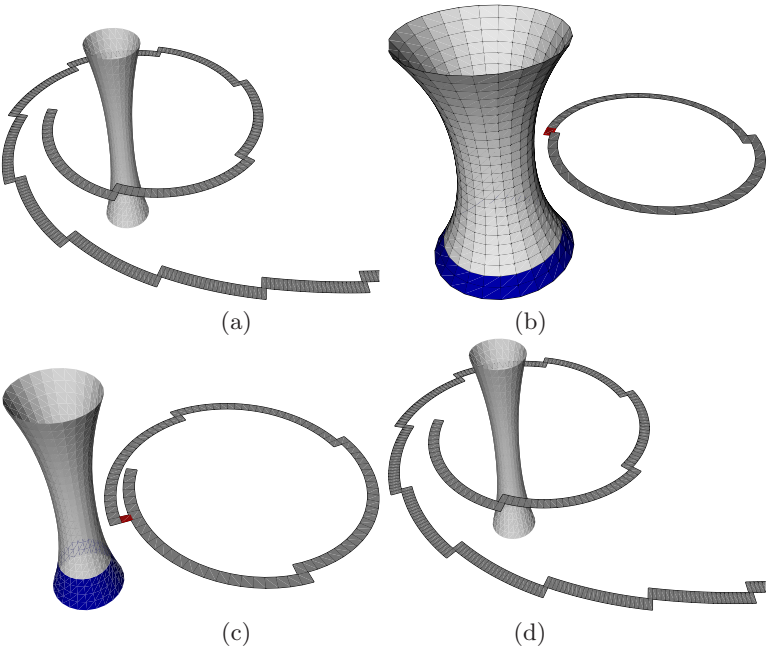


Fig. 5. Unfoldability is dependent on construction. Here k is fixed at 2.25, du and dv are fixed at 25, and $h = 1.2$, $h = 3.2$, $h = 6.2$ and $h = 9.2$. Only $h = 9.2$ is unfoldable.

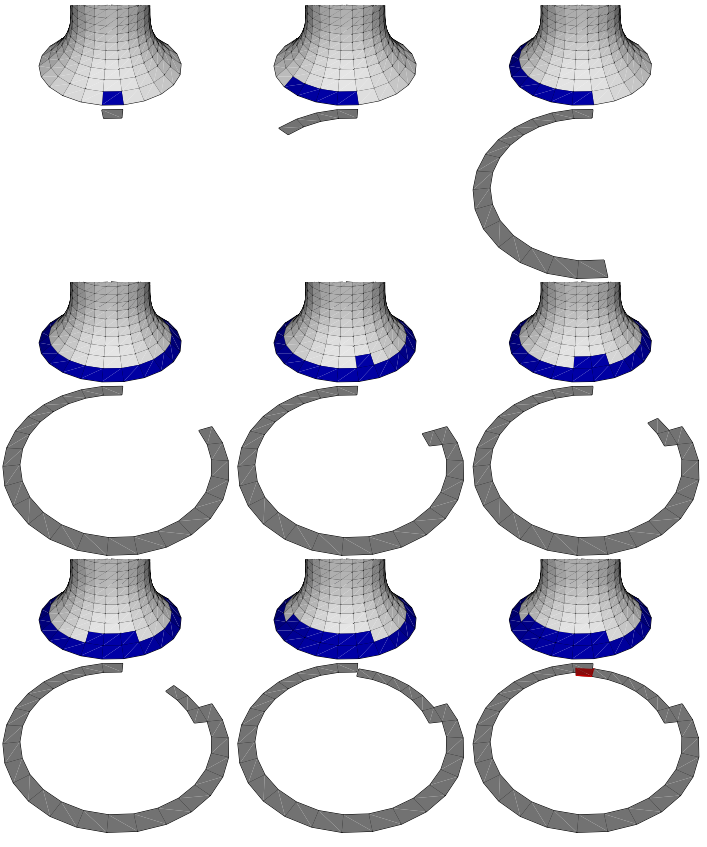


Fig. 6. Spiral unfolding with conflict in the second row

non-convex polyhedra, there exists no known method faster than exhaustive sampling for determining unfoldability.

Corollary 2 allowed restriction of development to linear stripping heuristics. A series of heuristic schemes were evaluated and tested on a varied set of coolinoid models. The heuristic ultimately chosen was a 'spiral' unfolding, which unravels the model into a strip of consecutive faces ordered by adjacency in a counter-clockwise traversal about the positive Y axis and then by ascending Y value. It was found that this scheme unfolded every model which could be unfolded by any other scheme plus a significant percentage more. No scheme was found which displayed a higher success rate.

Unwrapping a single row of faces from a coolinoid generates a curved arch of trapezoids which does not self-intersect; the flattened strip could never complete a full circle unless it already was one (which is the special case of the inner core of a torus, and undevelopable.) Each subsequent row, unrolled into a strip attached to the last face of the previous strip, will have less curve than the previous row

until the unwrapping crosses the vertical midplane, after which the curved arch of each row will wrap symmetrically in the opposite direction. This creates a smooth spiral which unravels to a straight line and then curls back into another spiral.

However, not all coolinoids can be spirally unfolded. The shared edge of the last face of a row i and the first face of row $i+1$ is the edge which lies *inside* the curl of the spiral; this moves each row inwards, towards the center of the spiral, by the height of the row. For many coolinoids this inwards step introduces a subsequent conflict between two rows, as shown in Figure 5a-c.

5.1 A Functional Expression of the Unfolding of a Coolinoid

The curve of the lower and upper borders of the spiral unfoldings of the faces of a coolinoid can be expressed as functions of a single linear parameter.

Taking h, k, du and dv as constants, define:

$$P(u,v) = \left[\left(cos\frac{2\pi u}{du} \right) \left(\left\| \frac{2v}{dv-1} \right\|^k + 1 \right), \frac{2vh}{dv-1}, \left(sin\frac{2\pi u}{du} \right) \left(\left\| \frac{2v}{dv-1} \right\|^k + 1 \right) \right]$$

$$Up(v) = P(0, v+1) - P(0, v)$$

$$Over(v) = P(1, v) - P(0, v)$$

$$\alpha(v) = cos^{-1} \left(\frac{Up(v)}{\|Up(v)\|} \cdot \frac{Over(v)}{\|Over(v)\|} \right)$$

$$\beta(v) = \pi - 2\alpha(v)$$

where

- $P(u,v)$ is the Coolinoid function. u, v range from 0 to $du-1$ and $dv-1$, respectively.
- $Up(v)$ is the step from the 'bottom' of row v to the 'top' of row v.
- $Over(v)$ is the step from one vertex at the bottom of row v to the next vertex in the row, ordering the vertices in a clockwise direction up the positive Y axis.
- $\alpha(v)$ is the acute inner angle of the trapezoidal faces of row v.
- $\beta(v)$ is the angle by which the unwrapping of row v will 'curl' in the plane at each face.

The following functions are then defined:

$$Turn(v) = \sum_{lvl=0}^{v} (du-1) * \beta[lvl]$$

$$Hop(v, \theta) = \|Up(v)\| \, [cos(\theta), sin(\theta)]$$

$$Skip(v,\theta) = \|Over(v)\|\,[cos(\theta), sin(\theta)]$$

$$Jump(u,v) = Hop(v-1, (Turn(v-1) + \alpha(v-1) + \beta(v-1))) + \sum_{i=1}^{u} Skip(v, Turn(v-1) + i * \beta(v))$$

$$Outer(u,v) = Jump(u,v) + \sum_{lvl=0}^{v-1} Jump(du-1, lvl)$$

$$Inner(u,v) = Outer(u,v) + Hop(v, Turn(v-1) + \alpha(v) + (u+1) * \beta(v)) + \lfloor u/(du-1) \rfloor * (Hop(v+1, Turn(v) + \alpha(v+1)) - Skip(v+1, Turn(v)))$$

such that

- $Turn(v)$ is the total curl introduced by the unwrapping of row v.
- $Hop(v,\theta)$ gives the vector $Up()$ rotated by θ.
- $Skip(v,\theta)$ gives the vector $Over()$ rotated by θ.
- $Jump(u,v)$ gives the offset of the unwrapping of the first u faces of row v.
- $Outer(u,v)$ give the position of the lower right corner of the unwrapping of the first $v-1$ rows and the first u faces of the v^{th} row.
- $Inner(u,v)$ gives the position of the upper right corner of the same faces as $Outer(u,v)$. The construction of $\lfloor \frac{u}{du-1} \rfloor$ is designed to shift the function back one face, allowing the first face of each row to share its lower edge with the last face of the row preceding.

The functions Outer() and Inner() are now defined over the range $\{u,v\}\ \varepsilon\ \{[0..du-1],[0..dv-1]\}$ with u and v both held integer. The unraveling unfolding evaluates these functions in a linear progression. Encapsulating this linear progression as

$$OuterRing(t) = Outer(\lfloor t \rfloor mod(du), \lfloor t/du \rfloor) \tag{1}$$

$$InnerRing(t) = Inner(\lfloor t \rfloor mod(du), \lfloor t/du \rfloor) \tag{2}$$

yields two univariate equations whose solution(s), if they exist, are the loci of intersection of the outer and inner border of the unfolding (Figure 7.) Testing for the unfoldability of a coolinoid is now reduced to solving for the intersection of these two equations.

6 Developability of the Coolinoid

As mesh resolution increases, the odds of the mesh being unfoldable decrease (Figure 8) echoing similar results gathered by J O'Rourke in Figure 2 of [O98]. The data shown in Figure 8 was gathered by taking the average developability over the range $\{h,k\}\varepsilon\{[0.5..14.75],[0..10]\}$ for each integer value of dim in the range [3..78], setting $du = dv = dim$.

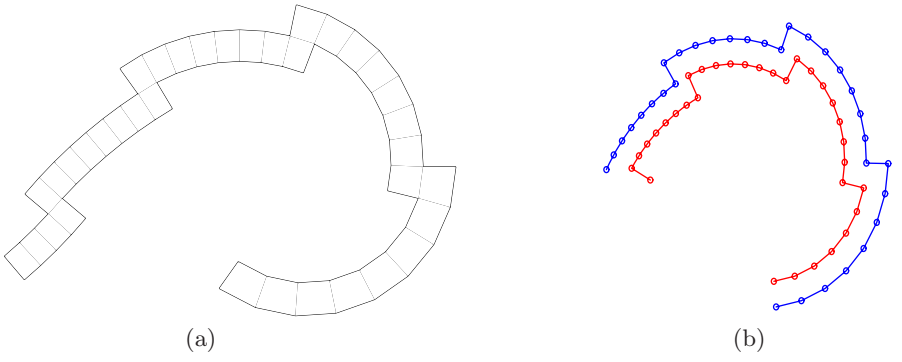


Fig. 7. (a) An unfolding (b) The same unfolding evaluated in Mathematica(tm)

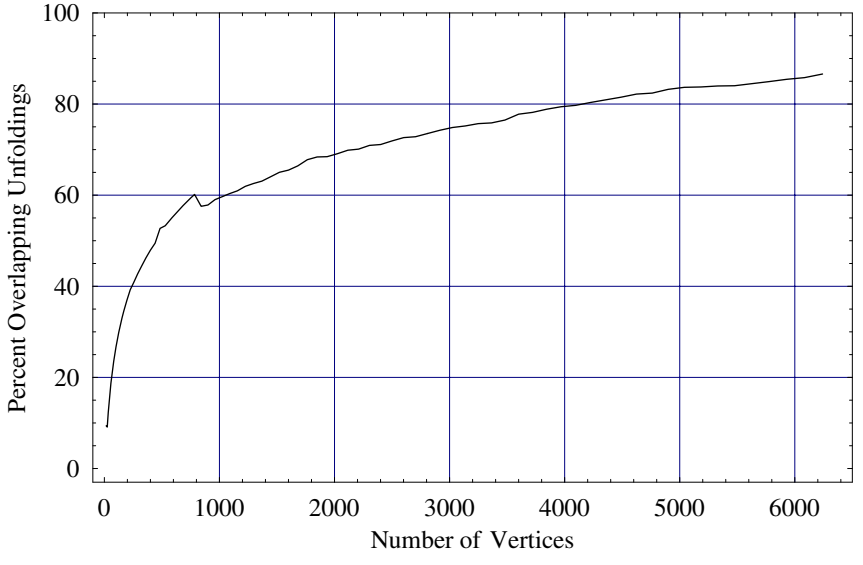


Fig. 8. Overlap vs Dimension

Evaluating the developability of coolinoids continuously over the domain given above, an implicit surface is generated (Figure 9.) The surface shows extremely intriguing behavior. Predictably, it displays interleaved shelving effects in some areas, highlighting regions of $\{h, k, dim\}$ space where parity (odd/even) of the mesh dimension has an effect on the developability; this is most readily visible where $k < 1$. Such liminal regions are common in such surfaces. More interestingly, an evolving wave pattern–hinting at fractal behavior–begins to appear in the isosurface as dimension increases. The wave hints at a much more complex isosurface than might be expected. Other ranges of symmetric and asymmetric behavior appear throughout the surface; further study is warranted.

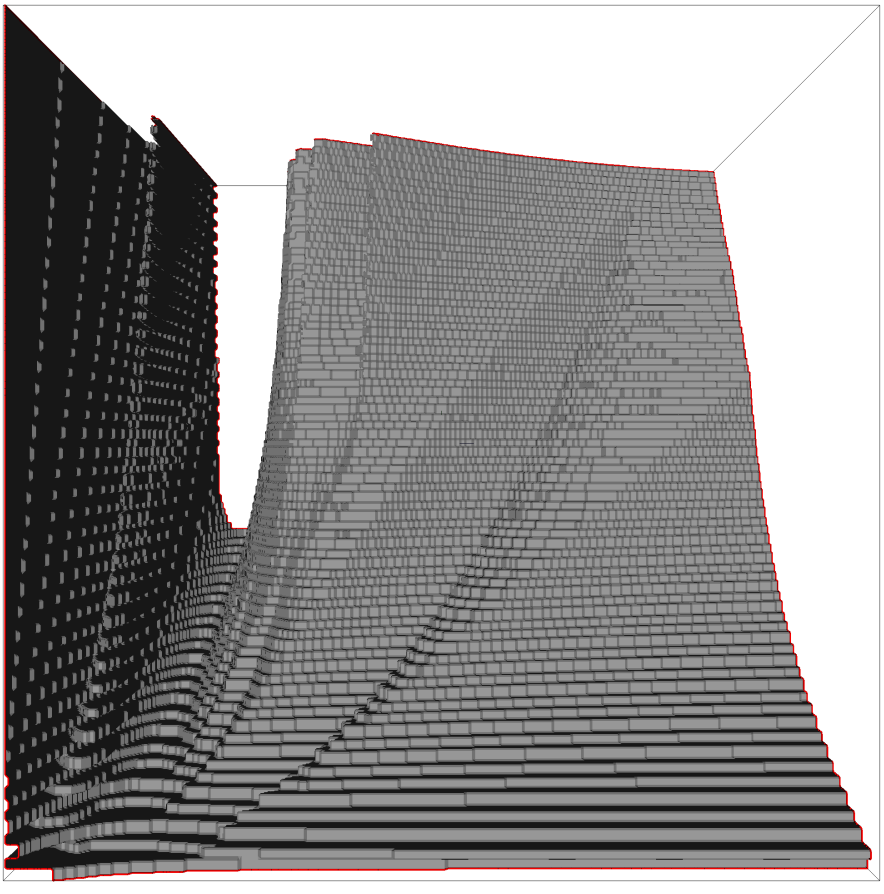


Fig. 9. Coolinoid unfoldability represented as an implicit surface in $\{h, k, dim\}$

In figure 9 the isosurface is shown looking from up the positive h axis. The positive k axis travels left-to-right and dim travels from the image's bottom to top. Note that the dim axis is integer, creating a voxel-like shelving effect (the 75 shelves in Figure 9) along the vertical axis.

Figure 10a shows the isosurface from the side, looking down the positive k axis towards the origin.

Figure 10b shows a detail of the lower h, k values. Note the interleaved parity-sensitive structure close to the origin in the $k \leq 1$ region, followed by a deep trough of undevelopability in the range $1 \leq k \leq 2$.

Figure 10c shows an overhead view, looking down on the model from the h axis. Note the wave pattern along the topmost border of the isosurface.

Figure 10d shows the model from above, looking directly down the positive dim axis. The wave in the isosurface is clearly visible.

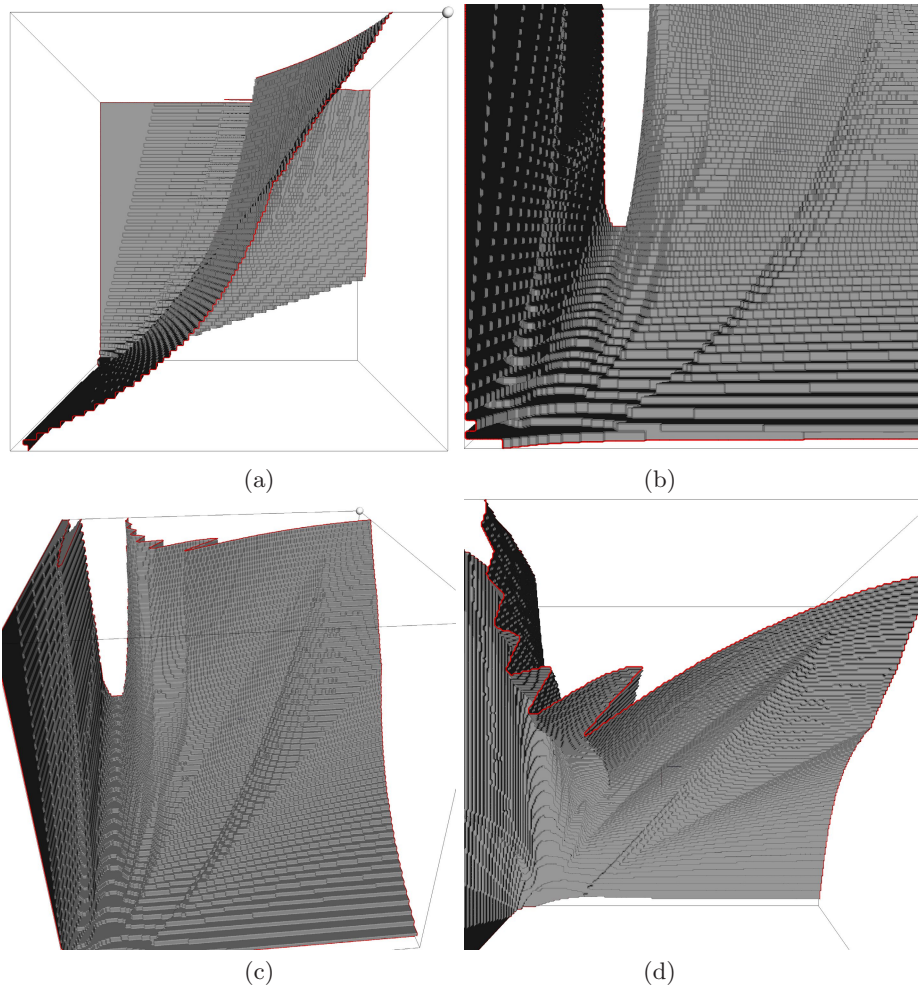


Fig. 10. Coolinoid unfoldability represented as an implicit surface in $\{h, k, dim\}$

7 Future Work

The coolinoid is very similar to functions such as $x^2 + y^2 - z^2 = 1$ and the *Catenoid*. Analysis comparable to that performed above would be quite instructive. The techniques demonstrated will apply to any surface which may be unfolded by the spiral unfolder.

A mathematically rigorous proof that the spiral unfolder was the optimal heuristic (ie., that there exists no coolinoid which is developable but which is not developed by the spiral unfolder) would be a significant advancement of the results presented.

The wave function which emerges in the upper ranges ($dim \rightarrow 50+$) of the coolinoid unfoldability isosurface displays fascinating fractal behavior which calls

for ongoing investigation. J O'Rourke has suggested[5] that the wave is an artifact of the stepwise nature of the integer *dim* field in conjunction with h, a progression already becoming visible in Figure 5 a-d.

The interleaving effects for low-resolution models (Figure 10b, lower left) decay as *dim* rises. Does that decay flatten fully, or is it re-expressed at higher resolutions in the much subtler interleaved effects that appear at higher values of h?

In the isosurface shown, it was assumed that $du = dv$. It would be very interesting to decouple these two fields, plotting a four-dimensional isosurface, substituting one of the four axes for time and animating the evolution of the wave.

8 Conclusions

It has been shown that simply-connected surfaces of negative interior curvature cannot be unfolded. An example has been given of a developable surface of negative interior curvature with two boundary curves: the *coolinoid*. A black-box solution for determining the unfoldability of any given coolinoid has been found and further analysis of the developability of the coolinoid has yielded startlingly complex and intriguing results.

Acknowledgements

The author gratefully acknowledges the guidance and advice of Dr. Malcolm Sabin, DAMTP, University of Cambridge.

References

[BCO04] Benbernou, N., Cahn, P., O'Rourke, J.: Unfolding Smooth Prismatoids. Smith College Computer Science Technical Report 078 (July 2004)

[BDEK99] Bern, M., Demaine, E., Eppstein, D., Kuo, E.: Ununfoldable Polyhedra. In: CCCG'99. Proc. 11th Canadian Conf. Computational Geometry, Vancouver, British Columbia, Canada, pp. 13–16 (August 15–18, 1999)

[G76] Griffiths, H.: Surfaces. Cambridge University Press, Cambridge ISBN: 0 521 20696 0

[MDSB03] Meyer, M., Desbrun, M., Schroder, P., Barr, A.: Discrete Differential Geometry Operators for Triangulated 2-Manifolds. In: VisMath '02 Proceedings (2002)

[NF94] Namiki, M., Fukuda, K.: Unfolding 3-dimensional Convex Polytopes: A Package for Mathematica 1.2 or 2.0., `ftp://ftp.ifor.math.ethz.ch/pub/fukuda/mathematica/UnfoldPolytope.tar.Z`

[O98] O'Rourke, J.: Folding and Unfolding in Computational Geometry. In: Akiyama, J., Kano, M., Urabe, M. (eds.) Discrete and Computational Geometry. LNCS, vol. 1763, pp. 258–266. Springer, Heidelberg (2000)

[5] Personal communications, February 2007.

[P03] Polthier, K.: Imaging maths - Unfolding Polyhedra, http://plus.maths.org/issue27/features/mathart/index.html

[S75] Shephard, G.: Convex Polytopes with Convex Nets. Math. Proc. Camb. Phil. Soc. (1975)

[S77] Strauss, W.: The Painter's Manual by Albrecht Durer (1525). Abaris Books (June 1977) ISBN: 0906212928

[V94] Van Loon, B.: Geodesic Domes. Tarquin Press (February 1994) ISBN: 0906212928

Rational Maximal Parametrisations of Dupin Cyclides

Helmut E. Bez

Department of Computer Science,
Loughborough University,
Leicestershire LE11 3TU, UK
h.e.bez@lboro.ac.uk

Abstract. Ring Dupin cyclides are algebraic surfaces of low degree that admit rational parametrisation. Their properties and applications in geometric modeling have been investigated in recent years by a number of authors. In particular their parametrisation using bi-quadratic Bézier patches is well documented. It is known, for example, that a minimum of four bi-quadratic Bézier patches is required to parametrise the entire manifold. However, neither the geometry nor the topology of the cyclide impedes the construction of single patch rational parametrisations of the whole surface. This paper constructs and discusses a number of single patch Bézier parametrisations of ring Dupin cyclides. Specifically: bi-quartic rational parametrisations of entire ring cyclides, optimized bi-quartic rational parametrisations of entire ring cyclides, and bi-sextic rational parametrisations of entire ring cyclides for which the parametrisation of iso-parametric lines is 'almost' identical to those of the familiar trigonometric parametrisation, are presented. The method of construction may be applied to the determination of rational patches of sub-maximal coverage, avoiding all the problems of other methods, such as: the complement of the intended region being parametrised, prohibited parametric values and other geometric constraints, and restriction of angular displacements to $< \pi$.

1 Introduction

The purpose of this paper is to consider maximal rational parametrisations of ring cyclides—where maximal means greatest possible coverage of the manifold by a single chart. The approach uses the trigonometric parametrisation, and various maximal, rational parametrisations of the circle S^1, to induce a number of rational, maximal parametrisations of cyclides. The construction is a natural exemplar of the path algebra approach of Bez & Wetzel [1,2,3] to inducing rational parametrisations of curves and surfaces from rational primitives. In particular, in this paper:

– rational maximal parametrisations of minimal degree are constructed on the parametric domain $(0,1) \times (0,1)$—these are bi-quartic and are denoted $\tau_{4,\lambda}$,

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 78–92, 2007.
© Springer-Verlag Berlin Heidelberg 2007

- Bézier parametrisations that are maximal and of minimal degree—compatible with also having positive weights, at least in the special case of the torus, are constructed—these are bi-quintic parametrisations and are denoted $\tau_{5,\lambda}$,
- optimized versions $\tau_{4,1+\sqrt{2}}$ and $\tau_{5,1+\sqrt{2}}$, of $\tau_{4,\lambda}$ and $\tau_{5,\lambda}$ are discussed,
- a maximal parametrisation with iso-parametric curves almost identical to those of the trigonometric parametrisation is constructed. It is bi-sextic and denoted τ_6.

2 Paths and Curves

If I is an interval of $\mathbb{R}$, including $I = \mathbb{R}$, then a regular path in $\mathbb{R}^n$ is a C^1 function $p : I \to \mathbb{R}^n$ with $p' \neq 0$ anywhere on I. A curve is an equivalence class of paths and $p \sim q$ if and only if there if there is a function $\phi : I_q \to I_p$ with $p = q \circ \phi$ – where I_p and I_q are the intervals of definition of p and q respectively. This definition actually refers to 1-manifolds immersed in $\mathbb{R}^n$ (i.e., curves in $\mathbb{R}^n$) and it is sufficient for our purposes. All reparametrisations of a path p may be expressed as $p \circ \phi$ for some reparametrisation function ϕ.

3 Rational Parametrisations of the Circle

In this paper the topology and parametrisation of $S^1 = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 - 1 = 0\}$ play a major role in the construction of parametrisations of cyclide surfaces; S^1 is a 1-dimensional differentiable manifold requiring a minimum of two co-ordinate charts for any atlas.

The maximum coverage of any chart, rational or otherwise, for S^1 is $S^1 \setminus \{\xi\}$ where ξ is any single point of S^1; the topology of S^1 does not permit a chart that covers the whole of S^1. The point ξ can be called the 'singular' point of the chart. Note that ξ is a singularity of the parametrisation and not of the manifold S^1.

Maximal quadratic rational charts exist for S^1, but they need to be defined on infinite domains, e.g. $(-\infty, \infty)$ [5]; i.e., no maximal rational quadratic chart for S^1 may be defined on the bounded interval $(0, 1)$. It is however possible to define a quadratic chart on $(0, 1)$ for S^1 that is 'almost' maximal, in the sense that given any $\varepsilon > 0$ there exists a quadratic parametrisation on $(0, 1)$ of S^1 for the sub-manifold $S^1 \setminus a_\varepsilon$, where a_ε is any arc of length ε. These parametrisations, although almost maximal, may not be used to induce maximal parametrisations of cyclides. They may however be used to induce almost maximal parametrisations of cyclides.

4 Optimized Maximal Bézier Parametrisations of S^1

This Section of the paper discusses the maximal rational parametrisations of S^1 used later in the paper to induce maximal rational parametrisations of cyclides.

4.1 Optimal Quartic Bézier Circles

By raising the degree to four, five and beyond, rational maximal parametrisations of increasingly uniform parametrisation can be constructed on $(0,1)$—see Bez & Wetzel [3] for full details. It should be noted that rational parametrisations of S^1 with exactly uniform parametrisation are not possible [7].

A family of degree 4 rational Bézier parametrisations, $\{c_{4,\lambda} : \lambda \geq 1\}$, of S^1 that generalise Chou's quartic [4] can be constructed. These parametrisations take the explicit form

$$c_{4,\lambda} = \frac{b_{0,4}(1,0) + \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{1,4}(0,1) + \frac{1}{6}\left(4 + \lambda^2 + \frac{1}{\lambda^2}\right) b_{2,4}(-1,0) + \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{3,4}(0,-1) + b_{4,4}(1,0)}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}}$$

where $\{b_{i,4}\}$ denotes the quartic Bézier basis and the weights are:

$$w_{0,4} = 1; \;\; w_{1,4} = 0; \;\; w_{2,4} = \frac{1}{6}\left(\lambda^2 + \frac{1}{\lambda^2}\right); \;\; w_{3,4} = 0; \;\; w_{4,4} = 1.$$

It should be noted that the vectors $(1,0),(0,1),(-1,0),(0,-1),(1,0)$ are not the vertices of $c_{4,\lambda}$, because the weights are not factored into the 'numerator'. This cannot be done without introducing vertices at ∞, due to the zero weights that occur, for $c_{4,\lambda}$.

For some reparametrisation function, $\psi_{4,\lambda} : (0,1) \to (0,1)$, the parametrisation $c_{4,\lambda}$ satisfies

$$p \circ \psi_{4,\lambda} = c_{4,\lambda};$$

i.e., we have:

$$\begin{aligned}
\cos 2\pi\psi_{4,\lambda} &= \left[\frac{b_{0,4}(1) + \frac{1}{6}\left(4 + \lambda^2 + \frac{1}{\lambda^2}\right) b_{2,4}(-1) + b_{4,4}(1)}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}}\right] \\
&= \frac{b_{0,4} - \frac{1}{6}\left(4 + \lambda^2 + \frac{1}{\lambda^2}\right) b_{2,4} + b_{4,4}}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}} \\
\sin 2\pi\psi_{4,\lambda} &= \left[\frac{\frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{1,4}(1) + \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{3,4}(-1)}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}}\right] \\
&= \frac{\frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{1,4} - \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{3,4}}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}}.
\end{aligned}$$

Chou's quartic corresponds to the choice $\lambda = 1$, and has the rate-of-tracing function shown on the left of Figure 1. More uniform parametrisations exist within the family and an optimum occurs at $\lambda = 1 + \sqrt{2}$. The rate-of-tracing function of this optimized circle $c_{4,1+\sqrt{2}}$ is shown on the right of Figure 1; this is much closer, both in an L^∞ and an L^2 sense, to the exactly-uniform rate-of-tracing function $\phi(t) = 2\pi$ than that of $c_{4,1}$.

Figure 2 shows the improvement in parametrisation, due to optimization, in an alternative way. The hollow circles show the exact constant-speed (or uniform) parametrisation, the parametrisations $c_{4,1}$ and $c_{4,(1+\sqrt{2})}$ are shown using filled circles.

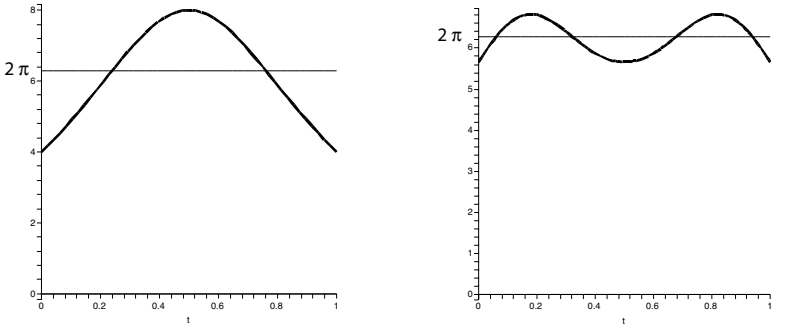


Fig. 1. Left: the 'bell' shaped rate-of-tracing function of the Chou quartic. Right: the graph of rate-of-tracing function of the circle $c_{4,1+\sqrt{2}}$.

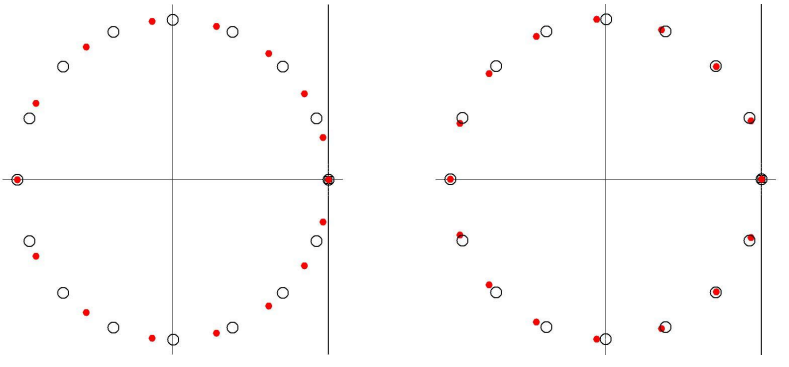


Fig. 2. Left: the rate-of-tracing of the Chou circle $c_{4,1}$ compared with the uniformly parametrized circle p. Right: the rate-of-tracing of the optimized circle $c_{4,1+\sqrt{2}}$ compared with the uniformly parametrized circle.

4.2 Optimal Quintic Bézier Circles

Chou [4] showed that if the degree of $c_{4,1}$ is raised to 5 then the weights of the quintic $c_{5,1}$ obtained in this way are all positive. The same applies to the parametrisations $c_{4,\lambda}$; for these the weights are:

$$w_{0,5} = 1; \quad w_{1,5} = \frac{1}{5}; \quad w_{2,5} = \frac{1}{10}\left(\lambda^2 + \frac{1}{\lambda^2}\right);$$

$$w_{3,5} = \frac{1}{10}\left(\lambda^2 + \frac{1}{\lambda^2}\right); \quad w_{4,5} = \frac{1}{5}; \quad w_{5,5} = 1.$$

Of particular interest here is the quintic $c_{5,1+\sqrt{2}}$ obtained by raising the degree of the optimized quartic $c_{4,1+\sqrt{2}}$ by one. The weights $\{w_{i,5}\}$ and vertices

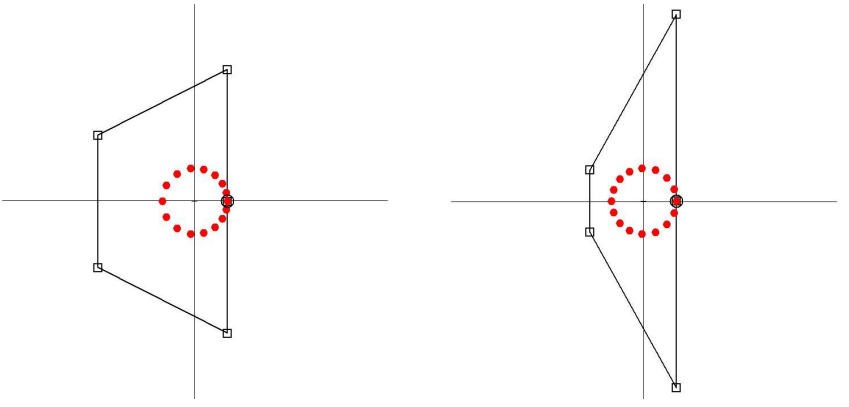


Fig. 3. Left: Bézier polygon for the Chou circle $c_{5,1}$. Right: Bézier polygon for the optimized circle $c_{5,1+\sqrt{2}}$.

$\{(x_{i,5}, y_{i,5})\}$ of $c_{5,1+\sqrt{2}}$ are given in the Appendix of this paper; a geometric comparison of the vertices of $c_{5,1}$ with those of the optimized circle $c_{5,1+\sqrt{2}}$ is given in Figure 3.

For some reparametrisation function $\psi_{5,1+\sqrt{2}} : (0,1) \rightarrow (0,1)$ the parametrisation $c_{5,1+\sqrt{2}}$ satisfies:

$$p \circ \psi_{5,1+\sqrt{2}} = c_{5,1+\sqrt{2}} = \frac{\sum_{i=0}^{5} b_{i,5}\ w_{i,5}\ (x_{i,5}, y_{i,5})}{\sum_{i=0}^{5} b_{i,5}\ w_{i,5}},$$

where $\{b_{i,5}\}$ denotes the quintic Bézier basis functions.

4.3 An Optimized Degree Six Rational Circle

An optimal, all-positive weight, degree 6 Bézier circle, denoted c_6, can be determined [3]. The parametrisation of c_6 is significantly more uniform than that of $c_{4,1+\sqrt{2}}$ and, as can be seen from Figure 4, is of 'almost' constant speed.

For some reparametrisation function $\psi_6 : (0,1) \rightarrow (0,1)$ the parametrisation c_6 satisfies:

$$p \circ \psi_6 = (\cos 2\pi\psi_6, \sin 2\pi\psi_6) = c_6 = \frac{\sum_{i=0}^{6} b_{i,6}\ w_{i,6}\ (x_{i,6}, y_{i,6})}{\sum_{i=0}^{6} b_{i,6}\ w_{i,6}}$$

where $\{b_{i,6}\}$ denotes the sextic Bézier basis functions, and weights $\{w_{i,6}\}$ and vertices $\{(x_{i,6}, y_{i,6})\}$ of c_6 are given in the Appendix of this paper.

Rational parametrizations of S^1 of degrees $8, 10, 12, \ldots$, of increasing closeness to the exact arc-length parametrisation, may also be constructed.

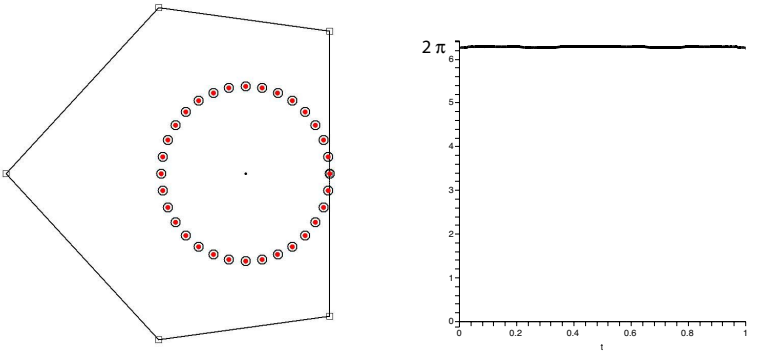


Fig. 4. Left: the optimal degree 6 Bézier circle c_6. Right: the graph of the rate-of-tracing function of c_6.

Fig. 5. Left: the surface D. Right: the surface D^*.

5 The Parametrisation of Ring Dupin Cyclides by Trigonometric Functions

Dupin cyclides may be defined implicitly as:

$$(x^2 + y^2 + z^2 - \mu^2 + b^2)^2 - 4(ax - c\mu)^2 - 4b^2y^2 = 0;$$

where the parameters a, b and c satisfy $c^2 = a^2 - b^2$, with $a \geq c$, and together with μ determine the cyclide type:

$c < \mu \leq a$ determines a ring cyclide,
$0 < \mu \leq c$ defines a horned cyclide,
$\mu > a$ corresponds to a spindle cyclide.

Only ring cyclides are true $2D$-manifolds. Figure 5 shows the manifold D for which $a = 6$, $b = 4\sqrt{2}$, $(c = 2)$ and $\mu = 3$.

A Dupin cyclide may be parametrised as $\tau_{tr} : [0, 1) \times [0, 1) \rightarrow D^*$ where

$$\tau_{tr,x}(\theta,\phi) = \frac{\mu(c - a\cos 2\pi\theta\cos 2\pi\phi) + b^2\cos 2\pi\theta}{a - c\cos 2\pi\theta\cos 2\pi\phi},$$
$$\tau_{tr,y}(\theta,\phi) = \frac{b(a - \mu\cos 2\pi\phi)\sin 2\pi\theta}{a - c\cos 2\pi\theta\cos 2\pi\phi},$$
$$\tau_{tr,z}(\theta,\phi) = \frac{b(c\cos 2\pi\theta - \mu)\sin 2\pi\phi}{a - c\cos 2\pi\theta\cos 2\pi\phi}.$$

In the interests of space efficiency we write this in homogeneous form as:

$$\tau_{tr}(\theta,\phi) = \begin{bmatrix} \mu(c - a\cos 2\pi\theta\cos 2\pi\phi) + b^2\cos 2\pi\theta \\ b(a - \mu\cos 2\pi\phi)\sin 2\pi\theta \\ b(c\cos 2\pi\theta - \mu)\sin 2\pi\phi \\ a - c\cos 2\pi\theta\cos 2\pi\phi \end{bmatrix};$$

τ_{tr} parametrises D except for a set of measure zero. We denote the region parametrised by τ_{tr} by D^*. The surface D^* with the un-parametrised region highly exaggerated is shown on the right of Figure 5.

The parametrisation τ_{tr} may not be extended beyond D^* due to discontinuities in σ^{-1} at points corresponding to parameter values $(\theta, 1)$, for all $0 \leq \theta < 1$, and $(1, \phi)$, for all $0 \leq \phi < 1$.

The parametrisation τ_{tr} is maximal in the sense that the topology of D does not permit parametrisations, rational or transcendental, of greater range. The rational parametrisations of D constructed in this paper parametrise precisely the same region of D as τ_{tr}.

6 Rational Maximal Parametrisations of D

Various rational, maximal parametrisations are discussed in the remaining sections of this paper. They are all obtained from τ_{tr} by reparametrisation; i.e., they all take the form

$$\tau_{tr} \circ \Psi$$

for some suitable reparametrisation function $\Psi : (0,1) \times (0,1) \to (0,1) \times (0,1)$; clearly all such reparametrisation functions may be expressed in the form

$$(\psi_\alpha(t,s), \psi_\beta(t,s)).$$

Cartesian product functions are a special case; in general if $f : X \to \Gamma$ and $g : Y \to \Omega$ then the Cartesian product function $f \times g : X \times Y \to \Gamma \times \Omega$ is defined by

$$(f \times g)(x,y) = (f(x), g(y)).$$

Clearly the products $g \times f$, $f \times f$ and $g \times g$ are all well-defined. For brevity we write $f^\times = f \times f$, $g^\times = g \times g$, etc.

All the induced rational parametrisations of D constructed below, by an inducing process, take the simplified generic homogeneous form:

$$\tau_{tr} \circ \psi^{\times} = \begin{bmatrix} \tau_{tr,x} \circ \psi^{\times} \\ \tau_{tr,y} \circ \psi^{\times} \\ \tau_{tr,z} \circ \psi^{\times} \\ \tau_{tr,w} \circ \psi^{\times} \end{bmatrix} = \begin{bmatrix} \mu(c - a\cos 2\pi\psi \cos 2\pi\psi) + b^2 \cos 2\pi\psi \\ b(a - \mu \cos 2\pi\psi) \sin 2\pi\psi \\ b(c \cos 2\pi\psi - \mu) \sin 2\pi\psi \\ a - c \cos 2\pi\psi \cos 2\pi\psi \end{bmatrix}$$

where $\psi^{\times}(t, s) = (\psi \times \psi)(t, s) = (\psi(t), \psi(s))$.

In this way D^* may be parametrised by single rational Bézier patches. The rational parametrisations of D, constructed later in the paper from τ_{tr}, parametrise the same region of D and have the same singular set as τ_{tr}.

7 Induced Parametrisations

Induced parametrisation provides an approach to the identification of suitable reparametrisation functions, Ψ, for the cyclide and other surfaces. The reparametrisation functions are not determined explicitly, but explicit forms for the rational functions they produce when composed with (products of) p are known.

Here we consider a particular class of surfaces, that includes the tori and the Dupin cyclides, for which rational parametrisations may be induced from a pair of circle parametrisations. The parametrisations of D constructed below use the Cartesian product $p \times p$ of two copies of p. Here $p(\theta) = (\cos(2\pi\theta), \sin(2\pi\theta))$ and $p \times p : (0, 1) \times (0, 1) \to \mathbb{R}^4$ is therefore:

$$(p \times p)(\theta, \phi) = (\cos(2\pi\theta), \sin(2\pi\theta), \cos(2\pi\phi), \sin(2\pi\phi)),$$

which we write as $p^{\times}$.

The Dupin cyclides are in the class of surfaces that may be parametrised using trigonometric functions in such a way that the parametrisation may be expressed as $F_M \circ p^{\times}$; where $F_M : \mathbb{R}^4 \to \mathbb{R}^4$ and composite function

$$F_M \circ p^{\times} : (0, 1) \times (0, 1) \to \mathbb{R}^4$$

is defined by

$$\begin{aligned} (F_M \circ p^{\times})(\theta, \phi) &= F_M(p^{\times}(\theta, \phi)) \\ &= F_M(\cos(2\pi\theta), \sin(2\pi\theta), \cos(2\pi\phi), \sin(2\pi\phi)). \end{aligned}$$

With F_D defined by

$$F_D(x, y, z, w) = \begin{bmatrix} \mu(c - axz) + b^2 x \\ by(a - \mu z) \\ bw(cx - \mu) \\ a - cxz \end{bmatrix}$$

it follows that $F_D \circ p^{\times}$ is the 'usual' trigonometric parametrisation of D.

Now, rational parametrisations for the circle S^1 can all be written as $p \circ \psi$, where $p = (\cos 2\pi\theta, \sin 2\pi\theta) = (p_1, p_2)(\theta)$, for some reparametrisation function $\psi : (a, b) \to (0, 1)$. Hence if a parametrisation of a particular surface may be written as $F_M \circ p^{\times}$, where

$$F_M : \mathbb{R}^4 \to \mathbb{R}^4$$

is rational in its four variables, then the rational parametrisation $p \circ \psi$ of p induces a rational parametrisation of the manifold M defined by $F_M \circ p^{\times}$. We have

$$F_M \circ (p \circ \psi)^{\times} : (a, b) \times (c, d) \to \mathbb{R}^4,$$

and

$$\begin{aligned} F_M \circ (p \circ \psi)^{\times}(t, s) &= F_M(p \circ \psi, p \circ \psi)(t, s), \\ &= F_M(\cos 2\pi\psi(t), \sin 2\pi\psi(t), \cos 2\pi\psi(s), \sin 2\pi\psi(s)); \end{aligned}$$

the function $F_D \circ (p \circ \psi)^{\times}$ is an induced rational parametrisation of the Dupin cyclide defined by the function F_D.

8 Induced Maximal Rational Parametrisations of Dupin Cyclides

8.1 Induced Maximal, Bi-quartic, Rational Parametrisation of D on $(0, 1) \times (0, 1)$

We have, from earlier in the paper, that

$$\begin{aligned} c_{4,\lambda} &= \left(\frac{b_{0,4} - \frac{1}{6}\left(4 + \lambda^2 + \frac{1}{\lambda^2}\right) b_{2,4} + b_{4,4}}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}}, \frac{\frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{1,4} - \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{3,4}}{b_{0,4} + w_{2,4} b_{2,4} + b_{4,4}} \right) \\ &= p \circ \psi_{4,\lambda} \\ &= (\cos 2\pi\psi_{4,\lambda}, \sin 2\pi\psi_{4,\lambda}) \end{aligned}$$

where $\{b_{i,4}\}$ denotes the quartic Bézier basis. The weights are:

$$w_{0,4} = 1; \;\; w_{1,4} = 0; \;\; w_{2,4} = \frac{1}{6}\left(\lambda^2 + \frac{1}{\lambda^2}\right); \;\; w_{3,4} = 0; \;\; w_{4,4} = 1.$$

The corresponding induced, bi-quartic, Bézier parametrisations of D is therefore:

$$\begin{aligned} \tau_{4,\lambda}(t, s) &= (\tau_{tr} \circ \psi_{4,\lambda}^{\times})(t, s) \\ &= \begin{bmatrix} \mu(c - a\cos 2\pi\psi_{4,\lambda}(t)\cos 2\pi\psi_{4,\lambda}(s)) + b^2 \cos 2\pi\psi_{4,\lambda}(t) \\ b(a - \mu\cos 2\pi\psi_{4,\lambda}(s))\sin 2\pi\psi_{4,\lambda}(t) \\ b(c\cos 2\pi\psi_{4,\lambda}(t) - \mu)\sin 2\pi\psi_{4,\lambda}(s) \\ a - c\cos 2\pi\psi_{4,\lambda}(t)\cos 2\pi\psi_{4,\lambda}(s) \end{bmatrix}. \end{aligned}$$

Fig. 6. Left: bi-quartic rational parametrisation $\tau_{4,1}$ of D induced by the 'standard' quartic Bézier circle $c_{4,1}$. Right: the more uniform rational bi-quartic parametrisation $\tau_{4,1+\sqrt{2}}$ of D induced by $c_{4,1+\sqrt{2}}$.

The substitutions

$$\cos 2\pi\psi_{4,\lambda}(t) = \frac{b_{0,4}(t) - \frac{1}{6}\left(4 + \lambda^2 + \frac{1}{\lambda^2}\right) b_{2,4}(t) + b_{4,4}(t)}{b_{0,4}(t) + w_{2,4} b_{2,4}(t) + b_{4,4}(t)},$$

$$\sin 2\pi\psi_{4,\lambda}(t) = \frac{\frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{1,4}(t) - \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{3,4}(t)}{b_{0,4}(t) + w_{2,4} b_{2,4}(t) + b_{4,4}(t)},$$

$$\cos 2\pi\psi_{4,\lambda}(s) = \frac{b_{0,4}(s) - \frac{1}{6}\left(4 + \lambda^2 + \frac{1}{\lambda^2}\right) b_{2,4}(s) + b_{4,4}(s)}{b_{0,4}(s) + w_{2,4} b_{2,4}(s) + b_{4,4}(s)},$$

$$\sin 2\pi\psi_{4,\lambda}(s) = \frac{\frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{1,4}(s) - \frac{1}{2}\left(\lambda + \frac{1}{\lambda}\right) b_{3,4}(s)}{b_{0,4}(s) + w_{2,4} b_{2,4}(s) + b_{4,4}(s)},$$

then produce the induced parametrisation of D^* on $(0,1) \times (0,1)$.

Figure 6 shows the iso-parametric curves of the parametrisation $\tau_{4,1}$ of D induced by $c_{4,1}$ and those of $\tau_{4,1+\sqrt{2}}$ induced by $c_{4,1+\sqrt{2}}$, where the more uniform nature of the optimized parametrisation $\tau_{4,1+\sqrt{2}}$ is apparent. Properties of $\tau_{4,1+\sqrt{2}}$ include:

- the parametrisation is of maximum range,
- the parametrisation is, at least visually, 'better' than the trigonometric parametrisation – compare Figure 6 (right) with Figure 7 (right).

Limitations include:

- parametrisations with positive weights, or at least non-zero weights, are preferred for many applications: these quartics have some zero weights and may not therefore be written in rational Bézier form, unless vertices at ∞ are allowed.

8.2 Induced Bi-quintic Maximal Rational Parametrisations of D on $(0,1) \times (0,1)$

The parametrisation of D induced by $c_{5,1+\sqrt{2}}$ has identical geometric properties to that induced by $c_{4,1+\sqrt{2}}$; however, at least in the special case of the torus,

the weights of the bi-quintic Bézier representation constructed in this way are all positive. We have

$$c_{5,1+\sqrt{2}} = \frac{\sum_{i=0}^{5} b_{i,5}\ w_{i,5}\ (x_{i,5}, y_{i,5})}{\sum_{i=0}^{5} b_{i,5}\ w_{i,5}} = p \circ \psi_{5,1+\sqrt{2}}$$

for some reparametrisation function $\psi_{5,1+\sqrt{2}}$. The corresponding bi-quintic, induced parametrisation of D is therefore:

$$\tau_{5,1+\sqrt{2}}(t,s) = (\tau_{tr} \circ \psi^{\times}_{5,1+\sqrt{2}})(t,s)$$

$$= \begin{bmatrix} \mu(c - a\cos 2\pi\psi_{5,1+\sqrt{2}}(t)\cos 2\pi\psi_{5,1+\sqrt{2}}(s)) + b^2\cos 2\pi\psi_{5,1+\sqrt{2}}(t) \\ b(a - \mu\cos 2\pi\psi_{5,1+\sqrt{2}}(s))\sin 2\pi\psi_{5,1+\sqrt{2}}(t) \\ b(c\cos 2\pi\psi_{5,1+\sqrt{2}}(t) - \mu)\sin 2\pi\psi_{5,1+\sqrt{2}}(s) \\ a - c\cos 2\pi\psi_{5,1+\sqrt{2}}(t)\cos 2\pi\psi_{5,1+\sqrt{2}}(s) \end{bmatrix}.$$

The substitutions

$$\cos 2\pi\psi_{5,1+\sqrt{2}}(t) = \frac{\sum_{i=0}^{5} b_{i,5}(t)\ w_{i,5}\ x_{i,5}}{\sum_{i=0}^{5} b_{i,5}(t)\ w_{i,5}},$$

$$\sin 2\pi\psi_{5,1+\sqrt{2}}(t) = \frac{\sum_{i=0}^{5} b_{i,5}(t)\ w_{i,5}\ y_{i,5}}{\sum_{i=0}^{5} b_{i,5}(t)\ w_{i,5}},$$

$$\cos 2\pi\psi_{5,1+\sqrt{2}}(s) = \frac{\sum_{i=0}^{5} b_{i,5}(s)\ w_{i,5}\ x_{i,5}}{\sum_{i=0}^{5} b_{i,5}(s)\ w_{i,5}},$$

$$\sin 2\pi\psi_{5,1+\sqrt{2}}(s) = \frac{\sum_{i=0}^{5} b_{i,5}(s)\ w_{i,5}\ y_{i,5}}{\sum_{i=0}^{5} b_{i,5}(s)\ w_{i,5}},$$

then produce the induced bi-quintic parametrisation of D^* on $(0,1)\times(0,1)$. The weights $\{w_{i,5}\}$ and vertices $\{(x_{i,5}, y_{i,5})\}$ are given in the Appendix of this paper.

Properties of $\tau_{5,1+\sqrt{2}}$ include:

– in the special case of the torus, the weights of $\tau_{5,1+\sqrt{2}}$ are all positive; this is the lowest degree, maximal parametrisation of the torus that is rational-Bézier with all positive weights. For the torus, the parametrisation may therefore be written in standard rational Bézier form with no vertices at ∞.

Limitations include:

– the parametrisation is identical to the quartic case—it therefore has the same geometric limitations as the bi-quartic parametrisation discussed above.

8.3 Induced Bi-sextic Maximal Parametrisation of D on $(0,1) \times (0,1)$ with Iso-parametric Curves 'Almost' Identical to Those of the Trigonometric Parametrisation

There exists a reparametrisation function ψ_6 such that the optimized degree-6 Bézier parametrisation c_6 of S^1 satisfies

$$p \circ \psi_6 = (\cos 2\pi\psi_6, \sin 2\pi\psi_6) = c_6 = \frac{\sum_{i=0}^{6} b_{i,6}\ w_{i,6}\ (x_{i,6}, y_{i,6})}{\sum_{i=0}^{6} b_{i,6}\ w_{i,6}}$$

where $\{b_{i,6} : 0 \leq i \leq 6\}$ are the degree-6 Bézier basis functions.

The corresponding induced bi-sextic parametrisation of D is:

$$\tau_6 = \tau_{tr} \circ \psi_6^{\times} = F_{\mathbb{T}} \circ p^{\times} \circ \psi_6^{\times}.$$

Explicitly we have:

$$\tau_6(t,s) = (\tau_{tr} \circ \psi_6^{\times})(t,s)$$

$$= \begin{bmatrix} \mu(c - a\cos 2\pi\psi_6(t)\cos 2\pi\psi_6(s)) + b^2\cos 2\pi\psi_6(t) \\ b(a - \mu\cos 2\pi\psi_6(s))\sin 2\pi\psi_6(t) \\ b(c\cos 2\pi\psi_6(t) - \mu)\sin 2\pi\psi_6(s) \\ a - c\cos 2\pi\psi_6(t)\cos 2\pi\psi_6(s) \end{bmatrix}.$$

The substitutions

$$\cos 2\pi\psi_6(t) = \frac{\sum_{i=0}^{6} b_{i,6}(t)\ w_{i,6}\ x_{i,6}}{\sum_{i=0}^{6} b_{i,6}(t)\ w_{i,6}},$$

$$\sin 2\pi\psi_6(t) = \frac{\sum_{i=0}^{6} b_{i,6}(t)\ w_{i,6}\ y_{i,6}}{\sum_{i=0}^{6} b_{i,6}(t)\ w_{i,6}},$$

$$\cos 2\pi\psi_6(s) = \frac{\sum_{i=0}^{6} b_{i,6}(s)\ w_{i,6}\ x_{i,6}}{\sum_{i=0}^{6} b_{i,6}(s)\ w_{i,6}},$$

$$\sin 2\pi\psi_6(s) = \frac{\sum_{i=0}^{6} b_{i,6}(s)\ w_{i,6}\ y_{i,6}}{\sum_{i=0}^{6} b_{i,6}(s)\ w_{i,6}},$$

using the weight and vertex data $\{w_{i,6}\}$, $\{(x_{i,6}, y_{i,6})\}$ from the Appendix, now produce a maximal bi-sextic parametrisation of D that is almost identical to the trigonometric parametrisation from which it is induced: see Figure 7. Figure 8 shows the relative least-squares difference between the trigonometric parametrisation τ_{tr} and the induced bi-sextic rational parametrisation τ_6, the maximum percentage difference being $\approx 0.12\%$.

Properties of τ_6 include:

- in the special case of the torus, the weights are positive,
- for all cyclides the parametrisation is very close to the trigonometric parametrisation.

Fig. 7. Left: bi-sextic parametrisation τ_6 of D induced from optimized degree-6 Bézier circles. Right: the standard trigonometric parametrisation τ_{tr} of D.

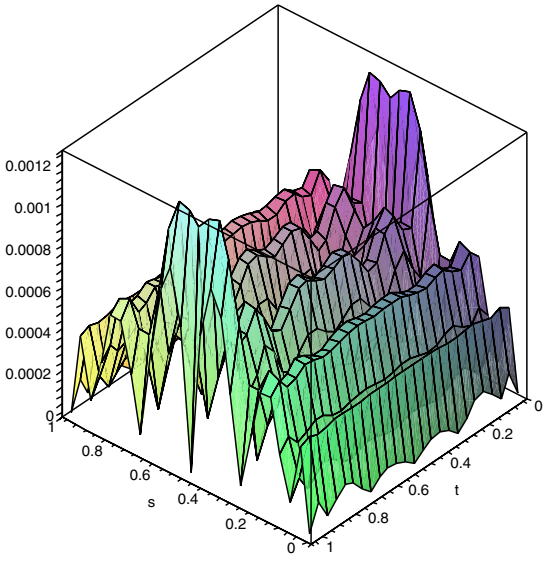


Fig. 8. The fractional differences between τ_{tr} and τ_6

Limitations include:

– possibly the 'high' degree of these parametrisations.

9 Sub-maximal Parametrisations of D

A slightly generalised procedure—inducing a parametrisation of D from two circular arcs—enables the determination of sub-maximal parametrisations of D. When applied to bi-quadratic parametrisations none of the usual problems of:

– complement of intended region being parametrised,
– prohibited parametric values and other geometric constraints,
– restriction of angular displacements to $< \pi$,

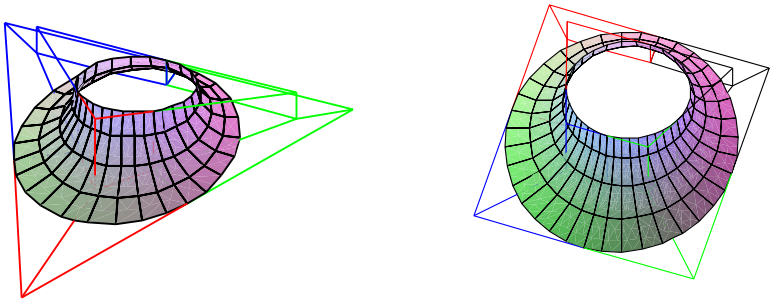


Fig. 9. 3- and 4- patch positive weight, bi-quadratic NURB representations a cylinder-plane blending cyclide

arise. The inducing construction reliably produces bi-quadratic patches on D—see Figure 9, which shows induced 3- and 4- patch, positive weight, bi-quadratic NURB constructions for a cylinder-plane blending cyclide.

References

1. Bez, H.E., Wetzel, T.J.: Constructive Path Algebra - a Tool for design, parametrisation and visualisation. In: Proceedings of Eurographics UK 2002, pp. 97–104. IEEE Computer Society Press, Los Alamitos (2002)
2. Bez, H.E., Wetzel, T.J.: Induced rational parametrisations of special curves. International Journal of Computer Mathematics 80, 1093–1109 (2003)
3. Bez, H.E., Wetzel, T.J.: Optimal rational circles of degrees five and Six. Journal for Geometry and Graphics 8, 1–16 (2004)
4. Chou, J.J.: Higher order Bézier circles. Computer-Aided Design 27, 303–309 (1995)
5. DeRose, T.: Rational Bézier curves and surfaces on projective domains. In: Farin, G. (ed.) NURBS for curve and surface design. SIAM, pp. 35–45 (1991)
6. Foufou, S., Garnier, L., Pratt, M.J.: Conversion of Dupin cyclide patches into rational bi-quadratic Bézier form. In: Martin, R., Bez, H., Sabin, M.A. (eds.) Mathematics of Surfaces XI. LNCS, vol. 3604, pp. 201–218. Springer, Heidelberg (2005)
7. Farouki, R.T., Sakkalis, T.: Real rational curves are not unit speed. Computer Aided Geometric Design 8, 151–157 (1991)
8. Martin, R., de Pont, J., Sharrock, T.J.: Cyclide surfaces in computer aided design. In: Proceedings IMA Mathematics of Surfaces, Oxford University Press, Oxford (1986)
9. Pratt, M.J.: Cyclides in computer aided geometric design. Computer Aided Geometric Design 7, 221–242 (1990)

Appendix: Weights and Vertices of the Optimized Circles

The weights and vertices of the optimized degree 5 Bézier circle are:

i	$w_{i,5}$	$v_{i,5} = (x_{i,5}, y_{i,5})$
0	1	$(1, 0)$
1	$\frac{1}{5}$	$(1, 4\sqrt{2})$
2	$\frac{3}{5}$	$(-\frac{5}{3}, \frac{2\sqrt{2}}{3})$
3	$\frac{3}{5}$	$(-\frac{5}{3}, -\frac{2\sqrt{2}}{3})$
4	$\frac{1}{5}$	$(1, -4\sqrt{2})$
5	1	$(1, 0)$.

The weights and vertices of the optimized degree 6 Bézier circle are:

i	$w_{i,6}$	$v_{i,6} = (x_{i,6}, y_{i,6})$
0	1.00000000	$(1.00000000, 0.00000000)$
1	0.63997907	$(1.00000000, 1.62906557)$
2	0.64192096	$(-1.03192823, 1.89695176)$
3	0.50963710	$(-2.83901798, 0.00000000)$
4	0.64192096	$(-1.03192823, -1.89695176)$
5	0.63997907	$(1.00000000, -1.62906557)$
6	1.00000000	$(1.00000000, 0.00000000)$.

Discrete Harmonic Functions from Local Coordinates

Tom Bobach[1], Gerald Farin[2], Dianne Hansford[2], and Georg Umlauf[1]

[1] CS Dept., University of Kaiserslautern, Germany
{bobach,umlauf}@informatik.uni-kl.de
[2] School of Computing and Informatics, Prism Lab., Arizona State University, USA
{gerald.farin,dianne.hansford}@asu.edu

Abstract. In this work we focus on approximations of continuous harmonic functions by discrete harmonic functions based on the discrete Laplacian in a triangulation of a point set. We show how the choice of edge weights based on generalized barycentric coordinates influences the approximation quality of discrete harmonic functions. Furthermore, we consider a varying point set to demonstrate that generalized barycentric coordinates based on natural neighbors admit discrete harmonic functions that continuously depend on the point set.

1 Introduction

Harmonic functions are defined by a vanishing Laplacian and can be computed as the unique solution of the Laplace equation with given boundary conditions [1]. Discrete harmonic functions are defined in a broader, abstract sense on general graphs [2]. If the nodes of the graph are points in the plane and the edges arise from a triangulation, the discrete harmonic function is a discretization of its continuous counterpart. The discrete Laplacian, characterizing a discrete harmonic function, is now defined as a linear combination of the function values at a point and at its one-ring neighbors in the triangulation. Given a point set, both the triangulation and the coefficients (edge weights) in the linear combination can be chosen freely, thus influencing the quality of the discrete approximation of harmonic functions.

There exist many approaches to triangulate point sets, driven by as many different objectives [3]. For points that have no further attributes than their position in space, the Delaunay triangulation is a widely adopted method to produce an almost unique triangulation whose properties are beneficial in may applications.

The definition of edge weights in an effort to discretize the Laplacian has received considerable attention in the past. In surface mesh processing and finite element methods, cotangent coordinates, also known as discrete harmonic coordinates, are a well-known method, see e.g. [4,5]. In the planar case, which is the focus of this paper, generalized barycentric coordinates of the points with respect to their one-ring neighborhood in the triangulation also provide good

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 93–103, 2007.
© Springer-Verlag Berlin Heidelberg 2007

approximations of the Laplacian. Generalized barycentric coordinates have been thoroughly studied for convex polygons [6], and have also been successfully derived for arbitrary polygons [7].

For point sets without explicit connectivity, the Voronoi diagram allows the definition of natural neighbor coordinates, which are generalized barycentric coordinates with various degrees of smoothness in an automatic neighborhood [8]. The duality of the Voronoi diagram and the Delaunay triangulation allows the comparison of natural neighbor coordinates to other generalized barycentric coordinates above. Three such natural neighbor coordinate methods are known as the C^0-smooth *Non-Sibsonian* coordinates [9,10,11], C^1-smooth *Sibson* coordinates [12], and C^2-smooth *Hiyoshi standard* coordinates [13].

Overview: We will start with the definition of discrete harmonic functions in Section 2, and revisit their computation in Section 3. In Section 4, we introduce six methods to compute generalized barycentric coordinates defined for polygons and arbitrary point sets in order to discretize the Laplacian. Section 5 compares discrete harmonic functions computed with the six discretizations in terms of approximation quality and suitability for spatially varying input data before concluding in Section 6.

2 Discrete Laplacian on Triangulations

A triangulation of a set of non-collinear points $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbf{R}^2$ is a partition of the convex hull $\mathcal{C}(X)$ into triangles $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ with edges $e_{ab}, e_{bc}, e_{ca} \in E$. The pair (X, E) is a directed, planar straight line graph with two directions on each non-boundary edge. The set of outgoing edges for a point $\mathbf{x}_i$ defines its one-ring neighborhood $N(\mathbf{x}_i) = \{\mathbf{x}_j | \exists e_{ij} \in E\}$.

To minimize the arbitrariness in the choice of E, we choose the Delaunay triangulation $\mathrm{Del}(X)$ such that $(X, E) = \mathrm{Del}(X)$. The circumcircle of a triangle in $\mathrm{Del}(X)$ does not contain any point from X in its interior, a property that uniquely and implicitly determines it up to ambiguous diagonals in the case of cocircular points [3].

Assume for now that we know weights $\lambda_{ij} \in \Lambda$, $\lambda_{ij} \geq 0$ for every edge e_{ij} and denote by $G = (X, E, \Lambda)$ the directed, weighted graph represented by the triangulation. Then, we consider scalar-valued functions $f_i := f(\mathbf{x}_i)$ on X and define the discrete Laplacian on G.

Definition 1 (Discrete Laplacian Δ). *On a connected, weighted graph* $G = (X, E, \Lambda)$ *we define the* discrete Laplacian *of* f *at* $\mathbf{x}_i$ *as*

$$\Delta f_i := \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} \lambda_{ij}(f_j - f_i). \tag{1}$$

The discrete Laplacian directly allows the definition of the local discrete harmonic property and discrete harmonic functions.

Definition 2 (Harmonic at a Point; Pole). *Let $G = (X, E, \Lambda)$ be a connected, weighted graph with $\lambda_{ij} \geq 0$. A function $f : X \to \mathbf{R}$ is called* harmonic *at node $\mathbf{x}_i$, if*

$$\Delta f_i = 0. \tag{2}$$

Otherwise f is said to have a pole *at $\mathbf{x}_i$.*

Definition 3 (Discrete Harmonic Function). *Denote by $P \subseteq X$ the set of poles for a function $f : X \to \mathbf{R}$, and by $H = X \setminus P$ the set of nodes at which f is harmonic. Then, f is called a* discrete harmonic function with poles P on G.

It is clear from the definition that each non-constant discrete harmonic function has at least two poles, i.e. at its minimum and maximum, since the function cannot be harmonic there. The later use of generalized barycentric coordinates motivates the identification of H with interior points, which are contained in the convex hull of their neighborhood, and of P with boundary points,

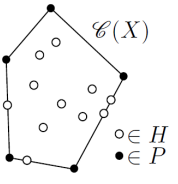


Fig. 1.

$$H := \{\mathbf{x}_i | \mathbf{x}_i \in \mathcal{C}(N(\mathbf{x}_i))\}, \qquad P := X \setminus H.$$

Note that there can be points in $H \cap \partial\mathcal{C}(X)$, see Figure 1. Finally, the following theorem ensures a unique solution for f based on given boundary conditions.

Theorem 1. *Let $G = (X, E, \Lambda)$ be a weighted, directed graph with $\lambda_{ij} \geq 0$. For every set $P = X \setminus H$, and every function $f|_P : P \to \mathbf{R}$, there is a unique function $f|_H : H \to \mathbf{R}$ such that $f|_X : X \to \mathbf{R}$ is a discrete harmonic function with poles P on G.*

For a proof, see [14].

3 Discrete Harmonic Functions

Recall that continuous harmonic functions are the unique solution to the Laplace equation on a domain Ω with boundary condition u,

$$\Delta f = 0, \qquad \text{s.t.} \quad f|_{\partial\Omega} = u.$$

Analogously, $f|_H$ is the unique discrete harmonic extension of $f|_P$ to f on G, i.e. $f|_H$ is the solution to the discrete Laplace equation with boundary condition $f|_P$,

$$\begin{aligned} \Delta f_i &= 0, && \text{for } \mathbf{x}_i \in H, \\ \text{s.t.} \quad f_i &= u(\mathbf{x}_i), && \text{for } \mathbf{x}_i \in P. \end{aligned}$$

Using the abbreviation $f_X = (f_H, f_P) \in \mathbf{R}^n$ for the column vector of f_i for all $\mathbf{x}_i \in X$ and analogously $f_H \in \mathbf{R}^m$, $f_P \in \mathbf{R}^{(n-m)}$, the Laplace operator Δf_H can be written, with appropriate reordering of X, as

$$\Delta f_H = f_H - S f_X = \begin{bmatrix} (\mathbf{I} - S^H) & S^P \end{bmatrix} \begin{bmatrix} f_H \\ f_P \end{bmatrix} = (\mathbf{I} - S^H) f_H + S^P f_P,$$

where $S \in \mathbf{R}^{m \times n}$ with $S_{ij} = \lambda_{ij}$ if $\mathbf{x}_i \in H$ and $\mathbf{x}_j \in V$, and $S_{ij} = 0$ otherwise, and $S = [S^H\ S^P]$ is the partition into $S^H \in \mathbf{R}^{m \times m}$ and $S^P \in \mathbf{R}^{m \times (n-m)}$. Setting $\Delta f_H = 0$ and solving for f_H yields

$$f_H = \underbrace{(\mathbf{I} - S^H)^{-1} S^P}_{=:\, M} f_P.$$

We see that M is only defined if $(\mathbf{I} - S^H)$ is invertible, which follows directly from Theorem 1 for positive weights.

Remark 1. The existence of M is also guaranteed by $\mathbf{I} - S^H$ being weakly diagonally dominant and irreducible. See [15] for a proof.

4 Local Coordinates

We have defined $\mathrm{Del}(X)$ and presented the computation of the discrete harmonic function f_H from f_P under the assumption that $\lambda_{ij} \geq 0$ are known. We now turn our attention to the computation of λ_{ij} from generalized barycentric coordinates.

By definition, $\mathbf{x}_i \in H$ can be expressed as a convex combination of $N(\mathbf{x}_i)$,

$$\mathbf{x}_i = \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} \lambda_{ij} \mathbf{x}_j, \qquad \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} \lambda_{ij} = 1, \qquad \lambda_{ij} \geq 0. \tag{3}$$

For $\mathbf{x}_i \in P$, there are in general no λ_{ij} with the above property. Since we do not need $\Delta_i, \mathbf{x}_i \in P$, we ignore them.

Obviously, property (2) of discrete harmonic functions directly corresponds to the *local coordinate property* (3), which we can rewrite as

$$0 = \Delta \mathbf{x}_i = \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} \lambda_{ij} (\mathbf{x}_j - \mathbf{x}_i), \qquad \mathbf{x}_i \in H. \tag{4}$$

Thus, the coordinate functions of $\mathbf{x}_i$ are harmonic functions on G with poles in P. The local coordinates λ_{ij} are coefficients in (4). If the positions in X change, their smoothness with respect to the positions in X governs the smoothness of the discrete Laplacian.

For $|N(\mathbf{x}_i)| > 3$, the λ_{ij} are not unique. We discuss the choice of local coordinates in the sequel.

4.1 Generalized Barycentric Coordinates in Polygons

In $\mathrm{Del}(X)$, each point $\mathbf{x}_i \in H$ is contained in the kernel of the simple polygon formed by the points of the oriented neighborhood $N(\mathbf{x}_i)$, which allows the computation of a wide spectrum of well-understood polygonal barycentric coordinates for $\mathbf{x}_i$. We consider cotangent- [5], Wachspress- [16], and mean value coordinates [17], which we refer to by Λ^c, Λ^w, and Λ^m, respectively. Note that Wachspress coordinates become negative in non-convex polygons.[1]

[1] In all our test cases, M was well defined.

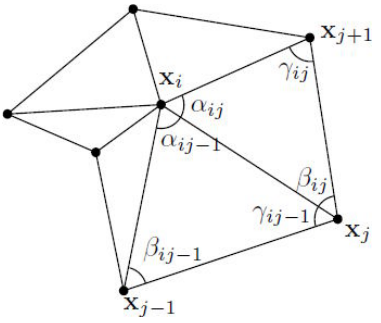


Fig. 2. Angles used in computations

The above are rational functions of the vertices of the enclosing polygon with linear precision on its edges. Polygonal barycentric coordinates smoothly depend on the vertices of the enclosing polygon, but in general do not agree if the enclosing polygon changes. This observation is important if the positions in X vary, as the neighborhood in $\mathrm{Del}(X)$ may change with X.

We now give the definition of the barycentric coordinates used in our assessment.

Definition 4 (Cotangent coordinates [5]). *Let β_{ij}, and γ_{ij} denote the angles as shown in Figure 2. Then,*

$$\lambda_{ij}^c := \frac{\tilde{\lambda}_{ij}^c}{\sum\limits_{\mathbf{x}_k \in N(\mathbf{x}_i)} \tilde{\lambda}_{ik}^c}, \qquad \tilde{\lambda}_{ij}^c = \cot(\beta_{ij-1}) + \cot(\gamma_{ij})$$

is called cotangent coordinate *of $\mathbf{x}_i$ with respect to $\mathbf{x}_j$.*

Definition 5 (Wachspress coordinates [16]). *Let β_{ij}, and γ_{ij} denote the angles as shown in Figure 2. Then,*

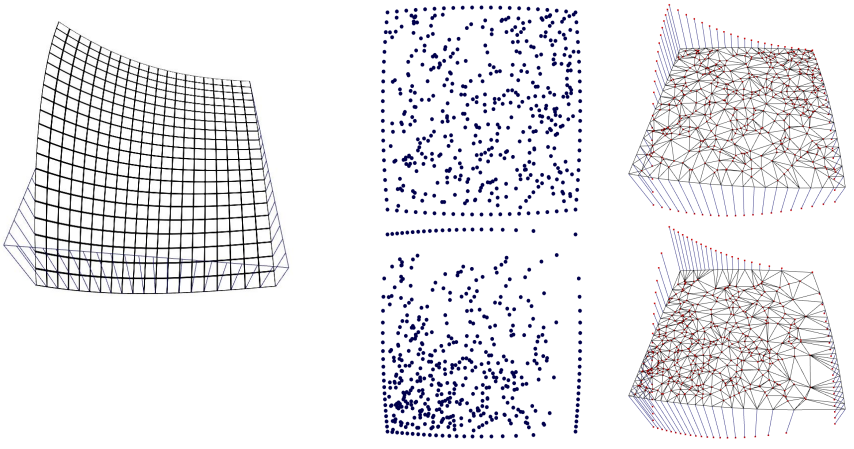
$$\lambda_{ij}^w = \frac{\tilde{\lambda}_{ij}^w}{\sum\limits_{\mathbf{x}_k \in N(\mathbf{x}_i)} \tilde{\lambda}_{ik}^w}, \qquad \tilde{\lambda}_{ij}^w = \frac{\cot(\beta_{ij}) + \cot(\gamma_{ij-1})}{\| \mathbf{x}_i - \mathbf{x}_j \|^2}$$

is called Wachspress coordinate *of $\mathbf{x}_i$ with respect to $\mathbf{x}_j$.*

Definition 6 (Mean value coordinates [17]). *Let α_{ij}, denote the angles as shown in Figure 2. Then,*

$$\lambda_{ij}^m = \frac{\tilde{\lambda}_{ij}^m}{\sum\limits_{\mathbf{x}_k \in N(\mathbf{x}_i)} \tilde{\lambda}_{ik}^m}, \qquad \tilde{\lambda}_{ij}^m = \frac{\tan(\alpha_{ij-1}/2) + \tan(\alpha_{ij}/2)}{\| \mathbf{x}_i - \mathbf{x}_j \|}$$

is called mean value coordinate *of $\mathbf{x}_i$ with respect to $\mathbf{x}_j$.*



(a) (b) (c)

Fig. 3. Homogeneous (upper row) and inhomogeneous (lower row) setting. (a) Graph of the function $g(x, y) = \frac{1}{3}x^3 - xy^2$, (b) point distribution, (c) Delaunay triangulation and values at the boundary.

4.2 Natural Neighbor Coordinates

Natural neighbor coordinates are generalized barycentric coordinates which are defined independent of any explicit adjacency on X, based on the implicit adjacency defined by the Voronoi diagram. However, the Delaunay triangulation is dual to the Voronoi diagram, and the set $N(\mathbf{x}_i)$ coincides with the set of natural neighbors upon which the coordinates are defined.

Three methods for the computation of natural neighbor coordinates are known as Non-Sibsonian- [10], Sibson- [12], and Hiyoshi standard natural neighbor coordinates [13], which we refer to by Λ^n, Λ^s, and Λ^h, respectively. They mainly differ in their smoothness with respect to $\mathbf{x}_i$.

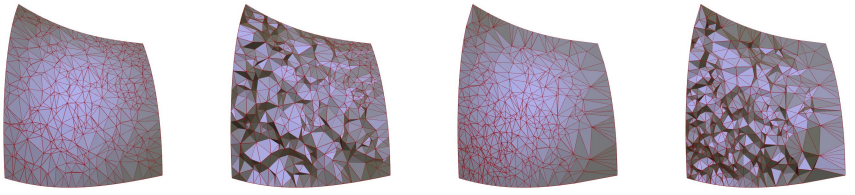
Before defining natural neighbor coordinates, we must introduce the Voronoi diagram $\mathrm{Vor}(X)$ as the dual of $\mathrm{Del}(X)$ in the sense that each edge e_{ij}^{Vor} is dual to e_{ij} and connects the circumcenters of two adjacent triangles. The polygon formed by the circumcenters of the Delaunay triangles adjacent to $\mathbf{x}_i$ is called its *Voronoi tile* $\mathcal{T}_i$.

Definition 7 (Non-Sibsonian coordinates [10]). *Let* $\| e_{ij} \|$ *and* $\| e_{ij}^{\mathrm{Vor}} \|$ *denote the lengths of* e_{ij} *and its dual* e_{ij}^{Vor}*, then*

$$\lambda_{ij}^n = \tilde{\lambda}_{ij}^n \;/\; \Big(\sum_{\mathbf{x}_k \in N(\mathbf{x}_i)} \tilde{\lambda}_{ik}^n \Big), \qquad \tilde{\lambda}_{ij}^n = \| e_{ij}^{\mathrm{Vor}} \| \;/\; \| e_{ij} \|$$

is called Non-Sibsonian coordinate *of* $\mathbf{x}_i$ *with respect to* $\mathbf{x}_j$*.*

Remark 2. Note that, given the same set of neighbors, cotangent coordinates are by construction identical to Non-Sibsonian coordinates, which is shown in [18].



(a) homogeneous, cotangent (b) homogeneous, inverse distance (c) inhomogeneous, cotangent (d) inhomogeneous, inverse distance

Fig. 4. Delaunay triangulation of X in 2.5D, the z-axis representing the values of f_i for the solution based on cotangent coordinates in (a), (c), and the solution based on inverse distance weights in (b), (d)

Consequently, cotangent coordinates on Delaunay triangulations belong to the family of natural neighbor coordinates.

Definition 8 (Sibson coordinates [12]). *Let* $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbf{R}^2$, *and* $X^{(i)} := X \setminus \{\mathbf{x}_i\}$. *Let* $\mathcal{T}_i$ *be the Voronoi tile of* $\mathbf{x}_i$ *in* $\mathrm{Vor}(X)$, *and* $\{\mathcal{T}_j^{(i)}\}_{\mathbf{x}_j \in N(\mathbf{x}_i)}$ *the Voronoi tiles of its neighbors in* $\mathrm{Vor}(X^{(i)})$. *Then, with* $|\cdot|$ *denoting the area in* $\mathbf{R}^2$,

$$\lambda_{ij}^s = |\mathcal{T}_i \cap \mathcal{T}_j^{(i)}| \;/\; |\mathcal{T}_i|$$

is called Sibson coordinate *of* $\mathbf{x}_i$ *with respect to* $\mathbf{x}_j$.

Hiyoshi standard coordinates emerge from a relationship by which one can express λ_{ij}^s as an integral over λ_{ij}^n in the context of power diagrams. The repeated application of this integral relation leads to a whole family of local coordinates with increasing order of smoothness. Because a complete definition of Hiyoshi standard coordinates does not fit the scope of this paper, we refer the reader to [13] and denote Hiyoshi C^2 (standard) coordinates by λ_{ij}^h.

Remark 3. In X, λ_{ij}^n are continuous, λ_{ij}^s are C^1-smooth, and λ_{ij}^h are C^2-smooth with respect to $\mathbf{x}_i$. Although the definitions above are not directly applicable for $\mathbf{x}_i \in H \cap \mathcal{C}(X)$, they can be extended accordingly by a limit argument. For $\mathbf{x}_i \in P$, this is not possible and λ_{ij}^l, λ_{ij}^s, λ_{ij}^h are not defined.

5 Comparison

In this section, we first discuss the experimental approximation quality of discrete harmonic functions based on one of the Laplacian discretizations presented in Section 4. Then, we illustrate the difference between generalized barycentric coordinates on polygons and natural neighbor coordinates in Laplacian discretizations on point sets with variable positions.

Table 1. Comparison of root mean square (RMS) error and maximum absolute deviation (MAX) for the per-point error $|f(\mathbf{x}_i) - g(\mathbf{x}_i)|$

Coordinates used for approximation	Homogeneous point distribution		Inhomogeneous point distribution	
	RMS	MAX	RMS	MAX
Non-Sibsonian coordinates	7.08e-8	1.08e-3	3.22e-7	3.88e-3
Sibson coordinates	2.53e-6	3.65e-3	1.92e-6	4.20e-3
Hiyoshi coordinates	5.78e-6	5.44e-3	3.57e-6	4.66e-3
Cotangent coordinates	7.08e-8	1.08e-3	3.22e-7	3.88e-3
Mean value coordinates	5.60e-6	4.85e-3	7.37e-6	7.04e-3
Wachspress coordinates	2.53e-5	1.10e-2	2.09e-5	1.18e-2
Inverse distance weights	5.43e-4	9.32e-2	8.76e-4	8.09e-2

5.1 Experimental Approximation Quality

We now compare discrete harmonic functions computed using the Laplacian discretizations based on coordinates from Section 4 by considering function values sampled from the harmonic function

$$g(x, y) = \frac{1}{3}x^3 - xy^2$$

over the domain $\Omega = [0.3, 1.3] \times [0.1, 1.1]$ to compute approximate solutions $f^\bullet$, $\bullet \in \{c, w, m, n, s, h\}$ to the Laplace equation with boundary conditions sampled from g. The graph of g is depicted in Figure 3(a). To illustrate the effect of a Laplacian discretization that does not arise from barycentric coordinates but still tries to capture the spatial relation between the nodes, our comparison includes inverse distance weights,

$$\lambda_{ij}^{id} := \|\mathbf{x}_i - \mathbf{x}_j\|^{-1} / \sum_{\mathbf{x}_k \in N(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{x}_k\|^{-1}.$$

We choose two distributions of $|X| = 480$ points in Ω: homogeneous as in Figure 3(b), top row, and inhomogeneous as in Figure 3(b), bottom row, which helps examine the influence of how well the domain is covered by sample points. In Figure 3(c) the Delaunay triangulation and the imposed boundary conditions are shown.

Table 1 shows the analysis of the per-point errors and indicates that Non-Sibsonian- and cotangent coordinates yield the lowest error, while the others are worse by orders of magnitude. Furthermore, it seems that this discrepancy is less prominent in the inhomogeneous setting for natural neighbor coordinates.

Figure 4 shows the visual comparison of f^c and f^{id}. The results obtained using any of the local coordinates in the Laplacian discretization are visually indistinguishable from the exact solution sampled from g (Figure 4(a) and (c)). As Figure 4(b) and (d) show, there are considerable distortions if the coordinate functions are not harmonic as with Λ^{id}.

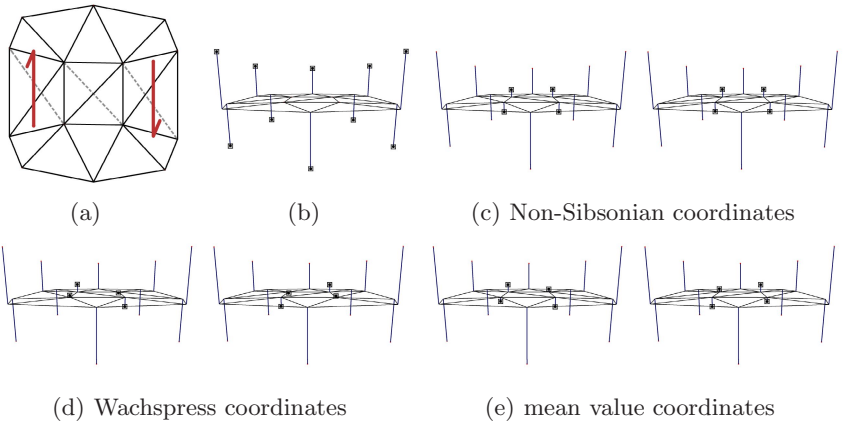


Fig. 5. (a) With the origin at the center of the data set and shear in the direction indicated by the arrows, the ambiguous Delaunay triangulation flips between two possible diagonal directions, depicted by the dashed lines. (b) The thick dots denote the values imposed as boundary condition, the interior four vertices are to be computed. (c)-(e) The thick dots indicate the computed discrete harmonic function values at the vertices of the triangulation. The left and right pictures show the values for the two different diagonal directions. Note the jumps in (d), (e) which does not occur in (c).

5.2 Varying Positions

We are now interested in how discrete harmonic functions based on the different Laplacian discretizations behave if the positions of the point set vary. Figure 5(a) shows $\mathrm{Del}(X)$ for a simple setting with $|H| = 4$, $|P| = 10$, which we deliberately choose to contain three sets of cocircular points. We assume the origin in the center of the dataset and shear the points in the direction indicated by the arrows, thus causing the ambiguous diagonals to flip between two possible positions as indicated by the dashed lines. Note that such flips occur in any triangulation if the deformation of the point set is big enough. Because the alternating boundary values shown in Figure 5(b) are horizontally symmetric, the discrete harmonic function should be, too.

Figure 5 shows the results of computing the discrete harmonic function on the point set under a shearing deformation, just before and after the diagonal flip. The results for cotangent, Non-Sibsonian, Sibson, and Hiyoshi standard coordinates are visually indistinguishable, which is why we only show one representative picture in Figure 5(c). First, notice how both Wachspress in Figure 5(d) and mean value coordinates in Figure 5(e) result in an asymmetric harmonic function in spite of the symmetry in the point set and the values. As long as the triangulation stays the same, the harmonic function for all approaches continuously follows the deformation. As a result of the diagonal flip, however, the polygonal coordinates show a discontinuous change in the harmonic function, which is not the case for natural neighbor coordinates.

6 Conclusion

In this paper, we have compared different discretizations of the Laplacian, applied to the computation of discrete harmonic functions from prescribed boundary values. To this end, we used the Delaunay triangulation and generalized barycentric coordinates in the Delaunay neighborhoods and focused on two classes of coordinates, defined on polygons and on natural neighbors.

First, we assessed the approximation quality by sampling the boundary values from a known harmonic function and comparing it to the computed discrete harmonic function. Although all approaches based on generalized barycentric coordinates provide acceptable results, we found that those based on Non-Sibsonian-coordinates, which are in our setting identical to cotangent coordinates, have the lowest approximation error and that by several orders of magnitude.

Second, we used a point set with fix boundary values but variable positions to analyze the effect of triangulation changes on the discrete harmonic function. We have showed that except for cotangent coordinates, those based on polygons yield discontinuous changes in the harmonic function under modifications in the triangulation. The reason for cotangent coordinates to behave differently lies in their equality to Non-Sibsonian coordinates if computed on a Delaunay triangulation, thus effectively representing natural neighbor coordinates. The latter always admit harmonic functions that continuously change with the positions of the point set, making them especially suited for application on variable point sets.

We are positive that within an appropriate framework these results also apply to the manifold case. One promising method is that of [19] which provides a setting similar to the one we used in this paper on manifolds.

Acknowledgments

This work was supported by the international graduate school DFG grant 1131 on ‘Visualization of Large and Unstructured Data Sets—Applications in Geospatial Planning, Modeling and Engineering’. Farin and Hansford were supported by an NSF grant 0306385 ‘Splines over Iterated Voronoi Diagrams’.

References

1. Axler, S., Bourdon, P., Ramey, W.: Harmonic Function Theory, 2nd edn. Springer, Heidelberg (2001)
2. Benjamini, I., Lovász, L.: Harmonic and analytic functions on graphs. J. of Geometry 76, 3–15 (2003)
3. Hjelle, Ø., Dæhlen, M.: Triangulations and Applications. In: Mathematics and Visualization. Springer, Heidelberg (2006)
4. Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W.: Multiresolution analysis of arbitrary meshes. Computer Graphics (Annual Conference Series) 29, 173–182 (1995)

5. Iserles, A.: A First Course in Numerical Analysis of Differential Equations. Cambridge University Press, Cambridge (1996)
6. Warren, J., Schaefer, S., Hirani, A.N., Desbrun, M.: Barycentric coordinates for convex sets. Advances in Computational Mathematics (to appear) (2005)
7. Hormann, K.: Barycentric coordinates for arbitrary polygons in the plane. Technical Report IfI-05-05, Department of Informatics, Clausthal University of Technology (February 2005)
8. Bobach, T., Umlauf, G.: Natural neighbor interpolation and order of continuity. In: GI Lecture Notes in Informatics: Visualization of Large and Unstructured Data Sets, pp. 69–86. Springer, Heidelberg (2006)
9. Christ, N.H., Friedberg, R., Lee, T.D.: Weights of links and plaquettes in a random lattice. Nuclear Physics B 210(3), 337–346 (1982)
10. Belikov, V., Ivanov, V., Kontorovich, V., Korytnik, S., Semenov, A.: The non-Sibsonian interpolation: A new method of interpolation of the values of a function on an arbitrary set of points. Comp. Mathematics and Mathematical Physics 37(1), 9–15 (1997)
11. Sugihara, K.: Surface interpolation based on new local coordinates. Computer Aided Design 13(1), 51–58 (1999)
12. Sibson, R.: A vector identity for the Dirichlet tessellation. Math. Proc. of Cambridge Philosophical Society 87, 151–155 (1980)
13. Hiyoshi, H., Sugihara, K.: Voronoi-based interpolation with higher continuity. In: Symposium on Comp. Geom, pp. 242–250 (2000)
14. Lovász, L.: Discrete analytic functions: a survey. Technical report, Microsoft Research (2000)
15. Varga, R.S.: Matrix Iterative Analysis. PrenticeHall, Englewood Cliffs, NJ, USA (1962)
16. Wachspress, E.: A rational finite element basis. Academic Press, San Diegoss (1975)
17. Floater, M.S.: Mean value coordinates. Comput. Aided Geom. Des. 20(1), 19–27 (2003)
18. Ju, T., Liepa, P., Warren, J.: A general geometric construction of coordinates in a convex simplicial polytope. Computer Aided Geometric Design 24, 161–178 (2007)
19. Flötotto, J.: A coordinate system associated to a point cloud issued from a manifold: definition, properties and applications. PhD thesis, Université de Nice-Sophia Antipolis (September 2003), `http://www.inria.fr/rrrt/tu-0805.html`

Computing the Topology of an Arrangement of Quartics

Jorge Caravantes and Laureano Gonzalez-Vega

Departamento de Matematicas, Estadistica y Computacion,
Universidad de Cantabria, Spain
jorge.caravantes@unican.es, laureano.gonzalez@unican.es

Abstract. We analyze how to compute in an efficient way the topology of an arrangement of quartic curves. We suggest a sweeping method that generalizes the one presented by Eigenwillig et al. for cubics. The proposed method avoids working with the roots of the involved resultants (most likely algebraic numbers) in order to give an exact and complete answer. We only treat in detail the cases of one and two curves because we do not introduce any significant variation in the several curves case with respect to Eigenwillig's paper.

1 Introduction

Efficient algoritms to study the topology of arrangements of real algebraic plane curves are available requiring, for moderate degrees of the polynomials defining the considered curves, the use of symbolic and algebraic methods. The case of a single curve has been widely studied: see[6] and more recently [7,11]. Efficient algorithms for arrangements of straight segments can be found in [5,9,10] and for conics in [2,12]. In this paper we adapt the method in [4], considering the case of cubic curves, to introduce a new algorithm for computing the topology of an arrangement of quartic curves.

The method in [4] for n cubics $f_1, \ldots, f_n$ finds the topology of each curve f_i, then it computes the topology of each pair f_i, f_j and, finally, it assembles all the available information. Since last step does not depend on the degree of the curves, it will be just sketched at the end of the paper.

To analyze a single curve $f(x, y) = 0$, we consider the roots $x_1 < x_2 < \ldots < x_n$ (that we will not determine explicitely) of the discriminant of f with respect to y. It is well known that the curve $f(x, y) = 0$ in the region $(x_j, x_{j+1}) \times \mathbb{R}$ consists (topologically) of a finite number of disjoint segments. The number of segments is exactly the number of real roots of $f(r_j, y)$ (at most 4 due to degree limitations) for any $r_j \in (x_j, x_{j+1})$. Now we have to find out what happens over each x_j. Since x_j is a real root of the discriminant of f (with respect to y), we know (or in other case we will be warned before) that $f(x_j, y)$ has exactly one multiple real root (representing what we call an event point of the curve) and up to two single real roots (representing what we will call uninvolved arcs). We

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 104–120, 2007.
© Springer-Verlag Berlin Heidelberg 2007

need to sort these roots (i.e. to find if the event is above, below or between the uninvolved arcs). The main differences with the cubic case is that in degree 3 there is at most one uninvolved arc. With this information we can construct a graph with the same topology of our curve.

For two curves $f(x,y)=0$ and $g(x,y)=0$, we consider as events both one–curve events and the intersections between them, so we sort their x–coordinates $x_1 < \ldots < x_n$. The idea is the same, first we find what happens over (x_j, x_{j+1}) with the help of a rational r_j. This time we have to sort the roots of $f(r_j,y)$ and $g(r_j,y)$ to find the relative positions of the arcs over the interval. For the events, we want over each x_j exactly one number y_j satisfying one (or various) of the following conditions:

- y_i is a multiple root of $f(x_j,y)$.
- y_i is a multiple root of $g(x_j,y)$.
- y_i is a common root of $f(x_j,y)$ and $g(x_j,y)$.

If this is satisfied then we sort the other roots of $f(x_j,y)$ and $g(x_j,y)$ and y_j, which represents the event now. When this is done, we have just to join the arcs in order to assemble the graph of the curves $f(x,y)=0$ and $g(x,y)=0$.

For the analysis of a whole arrangement, there is no additional modifications to include, so we refer to [4] with the obvious adaptations to quartics. The main contribution of this approach with respect to other methods is that we do not need to compute explicitely (or characterize) any of the x_i, so we will work just with rational numbers (and, ocasionally, with square roots).

The paper is organized as follows. After presenting some preliminaries in the first section about subresultants and special points of algebraic curves, the second section describes the method to analyze the topology of a single quartic. Third section explains the method for two quartics, showing the forth one how the proposed method works on a concrete example. We finish with a last section containing some conclusions and further research directions. We shall use freely in what follows notations and conventions from [4].

2 Preliminaries: Subresultants and Special Points of a Real Algebraic Plane Curve

Here we will introduce the main tools and notations we will use along this paper.

Definition 1. *Let*

$$P = \sum_{i=0}^{m} a_{m-i} x^i$$

and

$$Q = \sum_{i=0}^{n} b_{n-i} x^i$$

be two polynomials in x with coefficients in a field K. We define the j–th subresultant polynomial of P and Q with respect to the variable x in the following way (as in [8]):

$$\mathrm{Sres}_j(P,Q;x) = (-1)^{j(m-j+1)} \begin{vmatrix} a_0 & a_1 & a_2 & \cdots & \cdots & a_m & & & \\ & \ddots & \ddots & \ddots & & & \ddots & & \\ & & a_0 & a_1 & a_2 & \cdots & \cdots & a_m & \\ & & & & & 1 & -x & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & 1 & -x \\ b_0 & b_1 & b_2 & \cdots & \cdots & \cdots & b_n & & \\ & \ddots & \ddots & \ddots & & & & \ddots & \\ & & b_0 & b_1 & b_2 & \cdots & \cdots & \cdots & b_n \end{vmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \} n-j \\ \\ \\ \} j \\ \\ \\ \} m-j \end{matrix}$$

and we define the k–th subresultant coefficient of P and Q with respect to x, $\mathrm{sres}_k(P,Q;x)$, as the coefficient of x^k in $\mathrm{Sres}_k(P,Q,x)$. Finally, the resultant of P and Q with respect to x is:

$$\mathrm{res}(P,Q;x) = \mathrm{Sres}_0(P,Q;x) = \mathrm{sres}_0(P,Q;x) \ .$$

There are many ways of computing efficiently subresultants different from their definition: see, for example, [1] and the references contained therein. Next we introduce the main property of subresultants that will be used in what follows (see [1] for a proof).

Theorem 1. *Let f and g be two polynomials in y with coefficients in a field K. Then the following are equivalent:*

- $\mathrm{sres}_i(f,g;y) = 0$ *for all* $i < k$ *and* $\mathrm{sres}_k(f,g;y) \neq 0$.
- *The greatest common divisor of f and g has degree k and agrees with the polynomial* $\mathrm{Sres}_k(f,g;y)$ *(up to multiplication by nonzero elements of K).*

We now define some interesting points of a real algebraic plane curve implicitely defined which will guide the computation of the topological information we want to determine.

Definition 2. *Let f be a polynomial in two variables x and y with real coefficients. We write f_x and f_y for the partial derivatives of f and*

- $V(f) = \{(\alpha,\beta) \in \mathbb{R}^2 : f(\alpha,\beta) = 0\}$.
- $V_{\mathbb{C}}(f) = \{(\alpha,\beta) \in \mathbb{C}^2 : f(\alpha,\beta) = 0\}$.

Then:

- *A left (resp. right) x–extremal point of f is a point P in $V(f)$ for which all the branches of f through P are to the right (resp. left) of P (e.g. $(0,0)$ in $V(y^2 - x)$ is a left x–extremal point).*

- *A critical point is a point P in $V(f)$ such that $f_y(P) = 0$.*
- *A singularity is a point P in $V(f)$ such that $f_x(P) = f_y(P) = 0$.*
- *A node is an order two singularity through which exactly two branches of the curve pass:*
  - *if the two branches are real and the tangent lines to these branches at P are different then it is a crunode,*
  - *if the tangent lines coincide then it is a tacnode, and*
  - *if the two branches are imaginary (but the point is real) then it is an acnode.*
- *A cusp is an order two singularity with just one branch.*
- *A vertical flex is a critical point P such that $f_{yy}(P) = 0$ and $f_x(P) \neq 0$.*
- *An event or event point is a point P which is a singularity or an x–extremal point.*
- *Given an event point $P = (x_i, y_i)$, we say that an arc of $V(f)$ intersecting the line $V(x - x_i)$ is uninvolved if the intersection point between the arc and the line is different from P.*

Throughout this paper and when possible, we will abuse of notation identifying event points with their x–coordinates or, when fixed, with their y–coordinates (at those cases, there will be only one event point with such x or y coordinate).

The algorithm we are going to introduce to determine the topology of an arrangement of quartic curves is based upon a sweeping strategy where a vertical line will move from $-\infty$ to $+\infty$ detecting when the topology of the arrangement changes (i.e. at the event points: singularities or vertical tangency points).

Therefore we will use the fact that almost every point P of $V(f)$ (i.e. all but a finite number) is smooth and its tangent is not vertical. This means that we can assume, modulo a change of coordinates, that there are no two different events sharing the same x–coordinate. But also, by the implicit function theorem, this implies that there exists an analytic neighbourhood of P where $V(f)$ is the graph of an analytic function on the variable x (so, topologically, a segment). The only exceptions for this are the events.

Our sweeping algorithm will start by ordering the x–coordinates of the events $x_1 < \ldots < x_n$. Since all event points are over the x_i, all arcs over the open interval (x_i, x_{i+1}) are, topologically, parallel segments to the x–axis. So we can find rationals $r_1, \ldots, r_{n+1}$ such that $r_1 < x_1 < \ldots < x_n < r_{n+1}$. Then the data we need are the order of the real roots (with multiplicities) of every $f(x_i, y)$, always considering that $f(x_i, y)$ has an unique multiple root (the y–coordinate of the event) and the others are simple (the uninvolved arcs), and the number of real roots of each $f(r_i, y)$. With this, the simple roots of $f(x_i, y)$ belong to uninvolved arcs, so they are joined to exactly one root of $f(r_i, y)$ and one of $f(r_{i+1}, y)$. We join them with segments and when the uninvolved arcs are finished, the remaining points over the r_i are joined to the event points. In this way we obtain a graph which is topologically equivalent to $V(f)$. In the case of two (or more) curves we consider also as events the intersection points between them.

3 On the Topology of a Quartic Curve

We want our curve $f \in \mathbb{R}[x, y]$, $\deg(f) \leq 4$, to satisfy the following conditions:

- y–regularity (i.e. the coefficient of y^4 is a non zero constant),
- square-freeness (i.e. the polynomial f has no multiple factors),
- no two points of $V_{\mathbb{C}}(f) \cap V_{\mathbb{C}}(f_y)$ being covertical,
- no vertical flexes on f (i.e. every critical and non singular point of f is x–extremal), and
- no vertical singularities on f.

The algorithm to be presented in this section will produce the topology of the considered curve or detect infractions to these conditions if they happen. In case of an infraction, we perform a change of coordinates and try again (a finite number of times) because these infractions are known to only happen in a finite number of coordinate systems.

3.1 General Setup

We begin by checking y-regularity, which is easy. We then compute

$$R_f := \operatorname{res}(f, f_y; y)$$

and get its square-free decomposition

$$R_f = \prod_i (R_{f_i})^i \ .$$

If $R_f \equiv 0$ then f is not square-free, we replace f by its squarefree part:

$$\frac{f}{\gcd(f, f_y)}$$

and restart after performing a change of coordinates. Otherwise, we sort the real roots $x_1 < x_2 < \ldots < x_n$ of R_f together with their multiplicities $m_1, m_2, \ldots, m_n$. We now check the third condition:

> Two points of $V(f) \cap V(f_y)$ (whatever the field considered $\mathbb{R}$ or $\mathbb{C}$) are covertical if they share the same x–coordinate (let us say α, so our points are (α, β) and (α, β')). This means that $f(\alpha, y)$ and $f_y(\alpha, y)$ have two different common roots in the field we are working. Since the degree of f as a polynomial in y is 4 then it is impossible, due to conjugacy properties, to have simultaneously $\beta \in \mathbb{R}$ and $\beta' \notin \mathbb{R}$. So both are real or both are not (and in this latter case, they are conjugates). Anyway, if there are two covertical critical points, we have that
>
> $$\operatorname{sres}_1(f(\alpha, y), f_y(\alpha, y); y) = 0$$
>
> and
>
> $$\operatorname{res}(f_y(\alpha, y), f_{yy}(\alpha, y); y) \neq 0$$

(if both are equal to zero, there is a multiplicity three root, so coverticality is impossible since we work with quartics). So for the coverticality condition, we need all the roots of

$$\gcd(\mathrm{res}(f, f_y; y), \mathrm{sres}_1(f, f_y; y))$$

to be roots of $\mathrm{res}(f_y, f_{yy}; y)$. Otherwise we restart after performing a change of coordinates.

We now choose rational numbers r_i such that

$$r_1 < x_1 < r_2 < x_2 < \ldots < x_n < r_{n+1}$$

and determine the numbers k_i of different real roots of $f(r_i, y)$. We finish our set up by checking the forth condition:

Let x_i be a real root of R_f such that $f(x_i, y)$, $f_y(x_i, y)$ and $f_{yy}(x_i, y)$ have a common root y_i (that must be real since $f(x_i, y)$ has degree four and real coefficients, so the greatest common divisor of these three polynomials must be a polynomial of degree at most two). We would have a vertical flex if and only if y_i is not a root of $f_x(x_i, y)$. This means that $\mathrm{sres}_1(f, f_y; y)$ is zero in x_i. So we want $\mathrm{res}(\mathrm{Sres}_2(f, f_y; y), f_x; y)$ to vanish in x_i. Otherwise, we signal "vertical flex" and restart after performing a change of coordinates.

Now we have just to check the situation on every event point x–coordinate. If we have $m_i = 1$, then the intersection multiplicity of f and f_y at the event point is 1. Therefore f is smooth at the event and intersects f_y transversally at this event, so it can not be a vertical flex point (the intersection point would be tangential). Therefore, in this case ($m_i = 1$) we have a x–extremal point (smooth, its tangent is vertical and it is not a vertical flex). Otherwise, since we have avoided vertical flexes, we have a singularity. We now classify and analyze these events:

- x–extremal points,
- locatable singularities, and
- unlocatable singularities.

3.2 x–Extremal Points

Let x_i be a root of R_{f_1} (i.e. a root of R_f with multiplicity 1). Then $|k_i - k_{i+1}| = 2$. Depending on which k_j is bigger, we know if it comes from a left or right x–extremal point. Let us suppose $k_i < k_{i+1}$ (i.e. it comes from a left x-extremal point), since the other case is symmetric. The only possibilities (due to y–regularity) are $k_i = 0$ and $k_i = 2$. In the first case, there is no more information to know. For the second case, the only thing remaining to know is where the uninvolved arcs are located (over or under the event point).

We have that $f(x_i, y)$ has two simple and one double real roots. This means that there are two real roots of $f_{yy}(x_i, y)$ a and b. There are two possibilities:

1. Both $f(x_i, a)$ and $f(x_i, b)$ have different sign from $\text{lcoeff}(f; y)$. Then there is one uninvolved arc above the event and one below. In fact we can say the same for a and b instead of the event.
2. If there is at least one of the numbers $f(x_i, a)$ or $f(x_i, b)$ whose sign is equal to $\text{lcoeff}(f; y)$, then both arcs are above or both are below the event. There exist the possibility for a or b of being between the uninvolved arcs (that is when the signs of $f(x_i, a)$ and $f(x_i, b)$ are different), but the other one is for sure in the same relative position than the event.

And this situation does not change if we remain near the event to the left since we are dealing with a left x-extremal point (see Remark 1). Thus the required steps to determine the location of the uninvolved arcs are:

1. Choose $r_i' \in [r_i, x_i)$ so that $f_{yy}(r_i', y)$ has two real roots $a < b$ which are computed.
2. Compare the signs of $f(r_i', a)$ and $f(r_i', b)$ with the sign of $\text{lcoeff}(f; y)$:
   (a) If both are different from the sign of $\text{lcoeff}(f; y)$ then we have one arc above and one below.
   (b) If $\text{sign}(f(r_i', a)) = -\text{sign}(f(r_i', b)) = \text{sign}(\text{lcoeff}(f; y))$ (or $f(r_i', a) = 0$) then the uninvolved arcs are above the event.
   (c) If $\text{sign}(f(r_i', a)) = -\text{sign}(f(r_i', b)) = -\text{sign}(\text{lcoeff}(f; y))$ (or $f(r_i', b) = 0$) then the uninvolved arcs are below the event.
   (d) If $\text{sign}(f(r_i', a)) = \text{sign}(f(r_i', b)) = -\text{sign}(\text{lcoeff}(f; y))$ then we find how many real roots of $f(r_i', y)$ are in (b, ∞). If there are two then the uninvolved arcs are above. The other possibility is to have zero, and then the uninvolved arcs are below.

Remark 1. We choose a rational r_i' instead of working with x_i (that will be usually a real algebraic number) because we are just considering signs of the polynomials. This is because, by continuity, for a close enough r_i', the signs of $f(r_i', a)$, $f_{yy}(r_i', a)$, $f(r_i', b)$ and $f_{yy}(r_i', b)$ are the same as the signs of $f(x_i, a')$, $f_{yy}(x_i, a')$, $f(x_i, b')$ and $f_{yy}(x_i, b')$ respectively (where a' and b' are the roots of $f_y(x_i, y)$). This means that the graph of $z = f(r_i', y)$ will show the same signs, monotonies and convexities as the graph of $z = f(x_i, y)$, so we will know which is the relative extreme which contains the double root. Figures 1 and 2 show the graph of $z = f(r_i', y)$ for r_i' close enough to x_i (left) and the graph of $z = f(x_i, y)$ (right).

3.3 Locatable Singularities

Now we consider the case of x_i with $m_i > 1$ and $\deg R_{f_{m_i}} = 1$. In this case x_i is rational and so is the y–coordinate y_i (see [4]). So we shall work with the polynomial

$$\tilde{f}(x, y) \stackrel{\text{def}}{=} f(x - x_i, y - y_i)$$

in order to determine the type of the point (x_i, y_i).

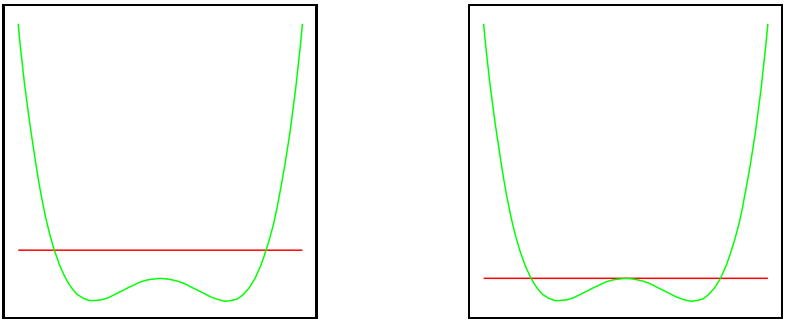


Fig. 1. Graph of $z = f(r'_i, y)$ (left) and of $z = f(x_i, y)$ (right) when $\text{sign}(f(r'_i, a)) = \text{sign}(f(r'_i, b)) \neq \text{sign}(\text{lcoeff}(f; y))$

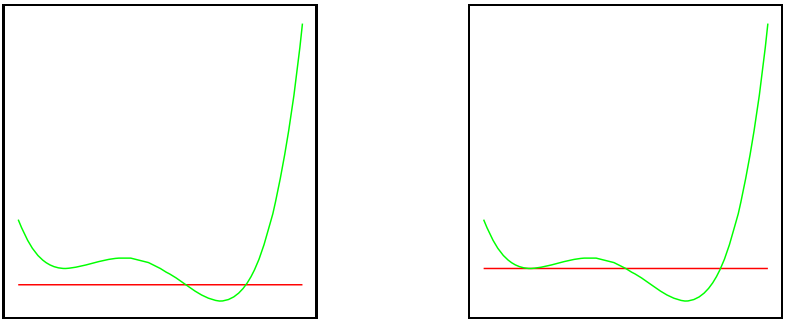


Fig. 2. Graph of $z = f(r'_i, y)$ (left) and of $z = f(x_i, y)$ (right) when $\text{sign}(f(r'_i, a)) = \text{sign}(f(r'_i, b)) = \text{sign}(\text{lcoeff}(f; y))$

First we consider its homogeneous decomposition

$$\tilde{f} = \tilde{f}_0 + \tilde{f}_1 + \tilde{f}_2 + \tilde{f}_3 + \tilde{f}_4.$$

We know that $\tilde{f}_0$ and $\tilde{f}_1$ are identically zero. Depending on the other two parts we have the following possibilities:

- If $\tilde{f}_2 = \tilde{f}_3 = 0$ then we have four lines meeting at a unique point.
- If $\tilde{f}_2 = 0 \neq \tilde{f}_3$ then we have a triple point. We find the unique non zero real root of $\tilde{f}(0, y)$ to find out if the uninvolved arc is the highest or the lowest one. All other arcs meet the singularity (from both sides). The reader is warned that this method just determines the topology at the singular point, so it will not distinguish between an ordinary triple point and a tacnode with another branch.
- If $\tilde{f}_2 \neq 0$ then we have a double point. We find where the uninvolved arcs are as described before and relate the other ones to the singularity.

3.4 Unlocatable Singularities

If $m_i > 1$ and $\deg R_{f_{m_i}} > 1$ then the real roots of $R_{f_{m_i}}$ are not so easy to determine as before. First of all, we check the value of $|k_i - k_{i+1}|$. We have a cusp if $|k_i - k_{i+1}| = 2$ and a node or a tacnode otherwise. In the cusp case, we work as in the case of x–extremal points (in fact it is a singular x–extremal point) in order to check where the uninvolved arcs are (if they exist).

To determine which type of singularity we have in the other case and the position of the uninvolved arcs, we proceed like in [4]. Let y_i be the double root of $f(x_i, y)$ and y_i' and y_i'' the single ones. We will define a polynomial $\delta(x)$ whose value in x_i will be $(y_i - y_i')(y_i - y_i'')$ in order to know if the event is between the arcs or not (i.e. above both or below both). We will also define a polynomial $\delta'(x)$ that will solve the above–below problem in the second case.

Consider the subresultant

$$\mathrm{Sres}_1(f, f_y; y) = \alpha(x)y + \beta(x) \ .$$

If $\alpha(x)$ has a common root $\tilde{x}$ with $R_{f_{m_i}}$, due to non coverticality, $f(\tilde{x}, y)$ would have a triple or quadruple root (so a triple or quadruple point, which would be locatable and then it is impossible, or a vertical singularity). Therefore if $\gcd(\mathrm{sres}_1(f, f_y, y), R_{f_{m_i}}) \neq 1$ then we signal "vertical singularity" and restart after performing a change of coordinates. Otherwise we use the extended euclidean algorithm to calculate the inverse $\gamma(x)$ of $\alpha(x)$ modulus $R_{f_{m_i}}$. From the properties of subresultants, we get the following equality for every root x_i of $R_{f_{m_i}}$:

$$\gamma(x_i)\beta(x_i) = y_i \ .$$

Therefore we can divide (considering y as main variable) f by

$$\mathrm{lcoeff}(f; y)(y - \gamma(x)\beta(x))^2$$

(we suggest for this using Horner's rule to divide f twice by $y - \gamma(x)\beta(x)$) since the remainder will vanish in every root of $R_{f_{m_i}}$. Let $H(x, y)$ be the quotient of this exact polynomial division. Obviously,

$$H(x_i, y) = (y - y_i')(y - y_i'')$$

for every root x_i of $R_{f_{m_i}}$. Then it is easy to conclude that we can define

$$\delta(x) := H(x, \gamma(x)\beta(x)) \ .$$

We define also

$$\delta'(x) := H_y(x, \gamma(x)\beta(x))$$

and $\delta_{\mathrm{disc}}(x)$ as the discriminant of $H(x, y)$ with respect to the variable y.

Once we have computed these polynomials, we check the positivity of them in the roots x_i of R_{f_2}, but first we determine the type of the involved singularity. For this we consider the following possibilities:

- If $k_i = k_{i+1} = 0$ then we have an acnode and there is no problem with the position of the uninvolved arcs since there are no such arcs.
- If $k_i = k_{i+1} = 4$ then we have a crunode and two uninvolved arcs (their position is to be studied below).
- If $k_i = k_{i+1} = 2$ then we evaluate $\delta_{\text{disc}}(x)$ on (or near enough to) x_i. If it is positive, this means that the single roots of $f(x_i, y)$ are real, so we have an acnode and two uninvolved arcs. If it is negative, the roots are complex and we have a crunode or acnode (they are topologically the same, so we do not go further) and no uninvolved arcs.

Now we look for the position of the uninvolved arcs: if $\delta(x_i)$ is negative then the event is between the uninvolved arcs. Otherwise, we check $\delta'(x_i)$: if it is positive then the uninvolved arcs are below the event and if negative, above. This is clear because, by construction,

$$\delta(x_i) = (y_i - y_i')(y_i - y_i'')$$

and

$$\delta'(x_i) = 2y_i - y_i' - y_i'' \ .$$

4 On the Topology of an Arrangement with Two Quartics

We analyze now the case of two curves f and g of degree 4. First of all, we list the conditions we want our pair of curves to satisfy (in addition to the conditions for a single curve) as in [4]:

- f and g are coprime.
- No two points of $V_{\mathbb{C}}(f) \cap V_{\mathbb{C}}(f_y)$ and $V_{\mathbb{C}}(g) \cap V_{\mathbb{C}}(g_y)$ are covertical.
- No point of $V_{\mathbb{R}}(f) \cap V_{\mathbb{R}}(g)$ is an x–extreme point of f or g.

Remark 2. As it is well known, all the conditions we are imposing can be avoided in general by a computation which is generally expensive in terms of efficiency. This algorithm, as before, computes the topology of the two curves or detects the infractions. In case of an infraction, we again perform a change of coordinates and try again, because these infractions are known to happen only a finite number of times.

4.1 General Setup

As in [4], we start by finding the x–coordinates of all event points. For this:

1. We perform the analysis of one curve for f and g (so we consider as already checked all the conditions relative to each curve alone).
2. We compute $R_{fg} := \text{res}(f, g; y)$. If $R_{fg} \equiv 0$ then we signal "not coprime", replace g by
$$\frac{g}{\gcd(f, g)}$$
and restart after performing a change of coordinates.

3. We determine the square–free decomposition of R_{fg}:

$$R_{fg} = \prod_j (R_{fg_j})^j \ .$$

4. We put together all the real roots of R_f, R_g and R_{fg} and order them:

$$x_1 < x_2 < \ldots < x_n,$$

considering their respective multiplicities $m_i^{(f)}$, $m_i^{(g)}$ and $m_i^{(fg)}$. If there exists $1 \leq i \leq n$ such that $0 < m_i^{(fg)} < m_i^{(f)} + m_i^{(g)}$ or $0 = m_i^{(fg)} < m_i^{(f)}, m_i^{(g)}$ then we signal "covertical events" and restart after performing a change of coordinates.
5. We choose $r_i \in \mathbb{Q}$ such that

$$r_1 < x_1 < r_2 < x_2 < \ldots < r_n < x_n < r_{n+1} \ .$$

6. We find and sort the real zeros of $f(r_i, y)$ and $g(r_i, y)$ to determine the sorted sequence of f–arcs and g–arcs over each interval between the events.

The only remaining thing is to analyze each event which is done in the rest of this section.

4.2 Checking Conditions on Intersection Points

In this subsection we introduce methods to check the coverticality conditions depending on the type of intersection.

Let x_i be the x–coordinate of an event with $m_i^{(fg)} > 1$ (i.e. an intersection point). If $m_i^{(f)} = m_i^{(g)} = 0$ then we need to prove that the multiplicity of the intersection is all in one point. For this we compute $\mathrm{sres}_1(f, g; y)$. If it vanishes in x_i then we have covertical events.

If $m_i^{(f)} = 1$ or $m_i^{(g)} = 1$ then we signal "x–extreme points over an intersection x–coordinate" and restart after performing a change of coordinates.

If $m_i^{(f)} \geq 2$ and $m_i^{(g)} = 0$ (or viceversa, but acting symmetrically) then we have to check that the intersection point is (exclusively) the singularity. If the singularity is locatable then we just take x_i and analyze $f(x_i, y)$ and $g(x_i, y)$. Otherwise:

- We consider $\mathrm{sres}_1(f, g; y)$. If it vanishes in x_i then we signal "covertical intersections" and restart after performing a change of coordinates. Now we know there is just one intersection between f and g over x_i. We now check that it is the right one.
- Take the first subresultant

$$p(x, y) = \mathrm{Sres}_j(f, f_y; y)$$

that does not vanish in x_i. Then $p(x_i, y) = (y - y_i)^j$ by Theorem 1. So if x_i is not a root of $\mathrm{res}(p, g; y)$ then we signal "covertical events" and restart after performing a change of coordinates.

If $m_i^{(f)} \geq m_i^{(g)} \geq 2$ then the process is obvious for locatable singularities. Otherwise:

– Take the first couple (j,k) such that that the subresultants

$$p(x,y) = \mathrm{Sres}_j(f, f_y; y)$$

and

$$q(x,y) = \mathrm{Sres}_k(g, g_y; y)$$

do not vanish in x_i. Then

$$p(x_i,y) = (y - y_{f,i})^j$$

(where $y_{f,i}$ is the y–coordinate of the event of f over x_i) and

$$q(x_i,y) = (y - y_{g,i})^k$$

(where $y_{g,i}$ is the y–coordinate of the event of g over x_i). If x_i is not a root of $\mathrm{res}(p,q;y)$ then, by Theorem 1, $y_{f,i} \neq y_{g,i}$, so we signal "covertical events" and restart after performing a change of coordinates. Now we know that our curves intersect each other in the singularity. Now we need to check that this is the only intersection point over x_i.

– Let $l = \min\{j,k\}$. If $\mathrm{sres}_{l+1}(f,g;y)$ is zero in x_i then this means that $(y-y_i)^{l+1}$ divides strictly $\gcd(f(x_i,y), g(x_i,y))$. So we signal "covertical intersections" and restart after performing a change of coordinates.

4.3 Intersections Singular–Singular

Since the coverticality is already checked, here we have nothing to do. We know the position of the uninvolved arcs from the one-curve analysis (this is an event for both curves), so there is nothing else to say.

4.4 At Least One of the Curves Is Regular at the Event Point

Let us suppose that the curve which is regular at the event point is g (if f is not regular, we already know the position of the event with respect to the uninvolved arcs of f; otherwise, we should work with f as we are working from now with g). Then $f(x_i,y)$ and $g(x_i,y)$ share y_i as the only common root. This means that if

$$\mathrm{Sres}_1(f(x,y), g(x,y); y) = \alpha(x)y + \beta(x),$$

then

$$y_i = -\frac{\beta(x_i)}{\alpha(x_i)}$$

(so, in particular, $\alpha(x_i) \neq 0$).

In this case we propose (to manage this case in its full generality) to work with rational functions. First of all we apply Horner's rule to divide g by $y + \beta(x)/\alpha(x)$

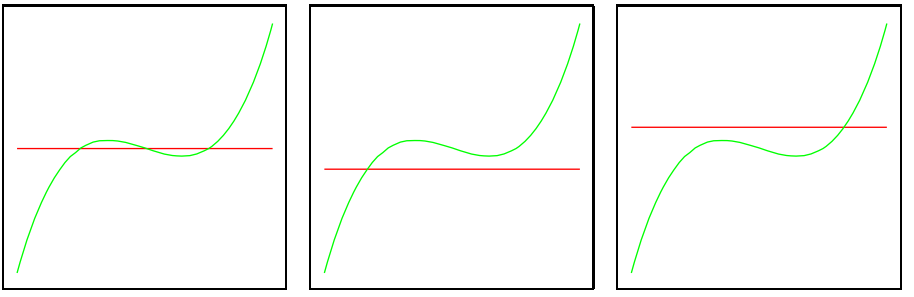


Fig. 3. Topological shape for the graph of $z = \tilde{g}(x_i, y)$ and the y–axis

and get a quotient $\tilde{g}$. The only thing that matters about the remainder is that it is zero in x_i, so if we specialize in such x–coordinate, the division will be exact and then

$$\tilde{g}(x_i, y) = \frac{g(x_i, y)}{y - y_i} \ .$$

Now let us suppose that the leading coefficient of g (which is y–regular) is positive. Then the graph of $\tilde{g}(x_i, y)$ and the y–axis have (topologically) one of the shapes in Fig. 3.

If g has exactly two arcs over x_i then we are in one of the last two cases, so $\tilde{g}(x_i, -\beta(x_i)/\alpha(x_i))$ is positive (it can not vanish) if and only if the event is in the upper arc (obviously, we test this positivity on a rational r close enough to x_i).

If we have four arcs of g over x_i then we are in the first case. We have the y–axes divided in four connected components. From left to right we have:

- one where $\tilde{g}(x_i, y)$ is negative, increasing and concave (second derivative is negative).
- one where $\tilde{g}(x_i, y)$ is positive and
  - increasing and concave (second derivative is negative); or
  - decreasing
- one where $\tilde{g}(x_i, y)$ is negative and
  - increasing and convex (second derivative is positive); or
  - decreasing.
- one where $\tilde{g}(x_i, y)$ is positive, increasing and convex (second derivative is positive).

So the only thing remaining is to check the signs of $\tilde{g}$ and its two first derivatives with respecto to y at $(r, -\beta(r)/\alpha(r))$ where r is a rational number close enough to x_i. And we will know the connected component y_i is in and then the position of the arc that contains the event in relation with the other three.

Remark 3. Note that, if one of the derivatives of $\tilde{g}(x_i, y)$ vanishes in y_i then y_i is in one of the middle intervals (so the sign of $\tilde{g}(x_i, y_i)$ provides all the needed information).

4.5 One Curve Events

Let us suppose the event is in f. If it is locatable then the analysis is trivial. Otherwise we have to consider two cases:

- the event is an acnode (i.e. the number of arcs of f to the left of the event, which we called k_i, is the same that the number of arcs to the right, k_{i+1}, and equal to the number of uninvolved arcs over the event); or
- at least, k_i or k_{i+1} is strictly bigger than the number of uninvolved arcs of f over x_i.

The second case is trivial since the event is not an isolated point and the order between the uninvolved arcs of f and g must be preserved since they do not take part in the event.

For the first case, we consider again the subresultant

$$\mathrm{Sres}_1(f, f_y; y)) = \alpha(x)y + \beta(x)$$

and we have

$$y_i = -\frac{\beta(x_i)}{\alpha(x_i)} \ .$$

There is an interval $[r_-, r_+]$ such that $x_i \in [r_-, r_+]$ and

$$g\left(x, -\frac{\beta(x)}{\alpha(x)}\right) \neq 0$$

for all $x \in [r_-, r_+]$. Then the relative position of y_i with respect to the real roots of $g(x_i, y)$ (i.e. the arcs of g over x_i) is the same of the relative position of $-\beta(r_+)/\alpha(r_+)$ with respect to the real roots of $g(r_+, y)$. And this can be checked easily by using for example Descartes' rule (see [1]).

5 An Example

In this section we apply the previously described algorithm to an easy example in order to show how it works. Consider the quartic curves

$$f := 75y - 92x^3 + 7y^4 + 74y^3x + 72y^2x^2 + 37yx^3$$

and

$$g := -8 - 29x^2 + 95y^3 + 12y^4 - 49y^2x^2 - 47yx^3 \ .$$

One can easily separate with Descartes' rule the real roots of the three resultants with the rationals $r_1 = -32$, $r_2 = -1/2$, $r_3 = 0$, $r_4 = 1$, $r_5 = 2$, and $r_6 = 32$. One also sees (using the corresponding squarefree decompositions of the three resultants) that the multiplicities of the roots are

- $m_1^{(f)} = m_2^{(f)} = 1$, $m_3^{(f)} = m_4^{(f)} = m_5^{(f)} = 0$,
- $m_1^{(g)} = m_2^{(g)} = m_3^{(g)} = 0$, $m_4^{(g)} = m_5^{(g)} = 1$, and
- $m_1^{(fg)} = m_2^{(fg)} = m_4^{(fg)} = m_5^{(fg)}$, and $m_3^{(fg)} = 1$.

Now we use again Descartes' rule to study the arcs over each r_i. So we separate the roots of $f(r_i, y)$ and $g(r_i, y)$ and find that the arcs over each r_i are (from below to above):

- for r_1: (g, g, f, f),
- for r_2: (g, f, f, g, f, f),
- for r_3: (g, f, f, g),
- for r_3: (f, g, f, g),
- for r_3: (f, g, g, g, f, g) and
- for r_6: (f, g, f, g).

Therefore, we have two x–extremal points for f, an intersection point between both curves and then two x–extremal points for g. First of all we analyze each curve separately. We illustrate in this example just how to proceed with the first x–extremal point of f, since the others are basically equal. We have two arcs of f over r_1 and four over r_2, so our x–extremal point is a left one. Then we have to choose a point r_1' in $[-32, x_1)$ such that no zeros of $\mathrm{res}(f_y, f_{yy}; y)$ or $\mathrm{res}(f_{yy}, f_{yyy}; y)$ lay in (r_1', x_1). We use again Descartes' rule to conclude that one possibility for this is $r_1 = -1$. Then we take the two roots $\alpha < \beta$ of $f(-1, y)$ and get that $f(-1, \alpha) < 0 < f(-1, \beta)$. Since the leading coefficient of f is positive, we get that the uninvolved arcs are above the event in this case.

Now we have to put together the arcs of f and g over x_1. We have to the left (g, g, f, f) and to the right (g, f, f, g, f, f) (where the two last arcs of f are the uninvolved ones). So, over x_i we have (from below to above): (g, "the event", g, f, f).

The cases of x_2, x_4 and x_5 are basically equal (or symmetric), so now we study the case of x_3. To find where the uninvolved arcs are we take

$$\begin{aligned}\mathrm{Sres}_1(f, g; y) = \alpha(x)y + \beta(x) = \\ = (33562896065 - 46145595768x + \ldots\ldots\ldots - 3032076974x^9)y \\ -45360000 + \ldots\ldots\ldots + 36307729635x^8 + 10161718228x^9.\end{aligned}$$

Now we compute the quotient $\tilde{g}$ of the euclidean divison of g by $y + \beta(x)/\alpha(x)$ and in the same way we get $\tilde{f}$. Using again Descartes' rule, one can find that $1/2 < x_3$ and that there are no roots of $\tilde{f}(x, -\beta(x)/\alpha(x))$ or $\tilde{g}(x, -\beta(x)/\alpha(x))$ in the interval $(1/2, x_3)$. Then we check the sign of $\tilde{f}(1/2, -\beta(1/2)/\alpha(1/2))$ (and the same for $\tilde{g}$) that is negative (and is exactly the same for $\tilde{g}$). Since the leading coefficients are 7 and 12, we have that the uninvolved arc of f is above the event (and the same for g).

Remark 4. This information can also be obtained by just checking that the lowest arc over r_3 belongs to g and the lowest one over r_4 belongs to f, but this method does not work for some tangential intersections and would force us to impose coverticality conditions that we are now able to avoid.

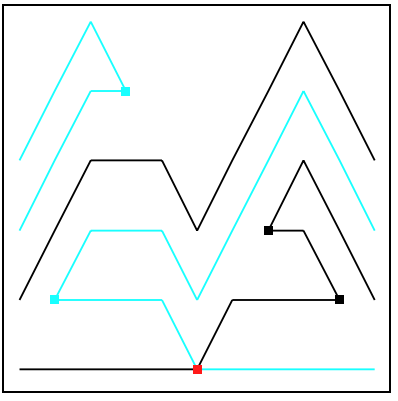


Fig. 4. Topology of the arrangement $f = 0$ (gray) and $g = 0$ (black): solid boxes represent the event points of the arrangement

Finally, we get over each x–coordinate of the considered points the following information:

		f						g		
	f	f	e				g	f	g	
f	f	g	g	g		g	f	g	f	g
f	g	f	f	f	g	f	e	g	g	f
g	e	f	f	f	f	g	g	g	e	g
g	g	g	g	g	e	f	f	f	f	f
r_1	x_1	r_2	x_2	r_3	x_3	r_4	x_4	r_5	x_5	r_6

where f or g represent a regular arc of f or g respectively, and we denote by e an event. Now we join the arcs and get the topology of the arrangement of the quartics f and g as shown in Fig. 4.

6 Conclusions

In order to compute the topology of an arrangement of quartic curves we have presented how to proceed when one or two quartics are to be analyzed. If there are more than two curves we follow very closely the method described in [4, Section 5] for cubics but replacing the one and two–curves analysis by the methods presented here. Note that the whole analysis rely on this two basic subtasks regarding the cases of one and two curves.

Let us suppose that $f_1, \ldots, f_r$ are the quartics whose arrangement is to be studied. Roughly, we will sort all the x–coordinates of the event points and choose rationals between them $r_1 < x_1 < r_2 < \ldots < x_n < r_{n+1}$. The relative position of the roots of the $f_j(r_i, y)$ is given by the different two curves analysis

performed together with the situation over the x_i (since an event involves one curve or, in other case two or more).

The whole algorithm has been implemented in `Maple` and the first obtained results are very promising. The performed experimentation will be analyzed and reported in [3].

Acknowledgments

This work was partially supported by the Spanish grant MTM2005-08690-C02-02 (Ministerio de Educacion y Ciencia).

References

1. Basu, S., Pollack, R., Roy, M.–F.: Algorithms in Real Algebraic Geometry. In: Algorithms and Computation in Mathematics, vol. 10, Springer, Heidelberg (2006)
2. Berberich, E., Eigenwillig, A., Hemmer, M., Hert, S., Mehlhorn, K., Schömer, E.: A Computational Basis for Conic Arcs and Boolean Operations on Conic Polygons. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 174–186. Springer, Heidelberg (2002)
3. Caravantes, J., Gonzalez–Vega, L.: Improving the Topology Computation of Arrangements of Low Degree Algebraic Curves. Work in progress (2007)
4. Eigenwillig, A., Kettner, L., Schömer, E., Wolpert, N.: Exact, Efficient and Complete Arrangement Computation for Cubic Curves. Computational Geometry 35, 36–73 (2006)
5. Flato, E., Halperin, D., Hanniel, I., Nechushtan, O., Ezra, E.: The Design and Implementation of Planar Maps in CGAL. ACM J. Experimental Algorithmics 5, article 13 (2000)
6. Gonzalez-Vega, L., Necula, I.: Efficient Topology Determination of Implicitly Defined Algebraic Plane Curves. Computer Aided Geometric Design 19, 719–743 (2002)
7. Eigenwillig, A., Kerber, M., Wolpert, N.: Fast and Exact Geometric Analysis of Real Algebraic Plane Curves. In: Proc. Int. Symp. Symbolic and Algebraic Computation (ISSAC 2007) (in print)
8. Li, Y.B.: A New Approach for Constructing Subresultants. Applied Math. and Computation 183, 471–476 (2006)
9. Mehlhorn, K., Noher, S.: LEDA: A Platform for Combinatorial and Geometric Computing. Cambridge University Press, Cambridge (1999)
10. Seel, M.: Implementation of Planar Nef Polyhedra. Report MPI-I-2001-1-003, Max-Planck-Institut für Informatik (2001)
11. Seidel, R., Wolpert, N.: On the Exact Computation of the Topology of Real Algebraic Curves (Exploiting a Little More Geometry and a Little Less Algebra). In: Proc. 21st Annual ACM Symp. Computational Geometry, pp. 107–115 (2005)
12. Wein, R.: High-level Filtering for Arrangements of Conic Arcs. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 884–895. Springer, Heidelberg (2002)
13. Wolpert, N.: Jacobi Curves: Computing the Exact Topology of Arrangements of Non-singular Algebraic Curves. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 532–543. Springer, Heidelberg (2003)

Non-uniform B-Spline Subdivision Using Refine and Smooth

Thomas J. Cashman[1], Neil A. Dodgson[1], and Malcolm A. Sabin[2]

[1] Computer Laboratory, University of Cambridge, CB3 0FD, England
tc270@cam.ac.uk
[2] Numerical Geometry Ltd, 26 Abbey Lane, Lode, Cambridge CB5 9EP, England

Abstract. Subdivision surfaces would be useful in a greater number of applications if an arbitrary-degree, non-uniform scheme existed that was a generalisation of NURBS. As a step towards building such a scheme, we investigate non-uniform analogues of the Lane-Riesenfeld 'refine and smooth' subdivision paradigm. We show that the assumptions made in constructing such an analogue are critical, and conclude that Schaefer's global knot insertion algorithm is the most promising route for further investigation in this area.

1 Background

NURBS have been an effective standard for representation of sculptured surfaces in manufacture since the late 1970's. Subdivision surfaces were also introduced at the end of the 1970's and have some significant advantages over NURBS. In particular they do not require a rigidly rectangular control grid. Both are based on the theory of B-splines and our current quad-grid subdivision surfaces can be looked on as a generalisation of a subset of NURBS.

The generalisation could be as useful in manufacture as it is in animation, where subdivision is now the de-facto standard. However, the aspects in which subdivision is only a subset are regarded as important to manufacture, and any new standard must be more compatible with the full generality of NURBS than current subdivision surfaces.

Our target is therefore a subdivision extension of NURBS in all their generality:

- non-uniform,
- rational, and
- of general degree.

Of these, the second is straightforward. Subdivision control points can be given weights and the process executed in projective space without any need for further theory.

Non-uniformity (unequal knot intervals) is not hard for low degree. Something more ambitious, assigning knot intervals to every edge on the control polyhedron, not just to every strip of polyhedron faces, was done by Sederberg et al. [10] for cubics and quadratics, and the linear case is trivial, but the handling of general degree is necessary for a subdivision method to be a generalisation of NURBS.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 121–137, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The work described here is an initial exploration of this target. We first address the univariate case, which we expect to map readily into tensor product surfaces. There is no point in making ad hoc rules for extraordinary vertices without having that firm foundation to build on.

In the uniform case, subdivision schemes for general degree B-splines are easily constructed and implemented using the Lane-Riesenfeld *refine and smooth* paradigm [7]. Instead of computing the new vertices directly by applying large stencils to the coarse polygon, a low degree refinement is first made locally, and then a sequence of smoothings by small, local filter operations is applied. The more smoothing steps, the higher the degree of the limit curve.

In the bivariate case, this approach gives efficiency as well as cleanness of code. Applying k 2×2 smoothing filters uses $4k$ multiplications per coordinate per new vertex; applying $\frac{1}{2}k$ 3×3 filters uses $4\frac{1}{2}k$. Applying a single $k \times k$ filter takes k^2 which is significantly more for large k.

The advantage is felt even more keenly when extraordinary vertices are introduced. Designing high-degree stencils for each valency is a non-trivial task, and designing rules for all possible combinations of extraordinary points within a large stencil is not a sensible ambition.

Although we are not addressing extraordinary points yet, the knowledge that we shall be looking at it in the future helps to steer the early work.

We want to formulate the refinement process for general degree non-uniform B-spline subdivision in terms of a composite of steps using small stencils. The archetype is the uniform refine and smooth approach.

We know exactly the complete subdivision matrices for general degree and general knot vectors. Once it has been decided what new knots are to be inserted (and this is a separate issue which we plan to address systematically), determining the new polygon is just a particular case of knot insertion. The algorithm used to insert the new knots may affect our view of the problem but is otherwise irrelevant, as the subdivision matrix is unique.

The question is how that wide-band matrix can be factorized into the narrow-band factors which correspond to the small-stencil operations that we need. This is the question addressed by the remainder of this paper.

1.1 Contents

After establishing some notation we start this exploration in Section 2, by looking at the matrices for order k and $k+2$ in the middle of a long polygon with a non-uniform knot-vector. This approach fails to find a factorisation in the general case, so we consider the particular case of Bézier end conditions in Section 3. This sub-problem is particularly important, because current subdivision schemes make a choice between

- using control points outside the surface (which leads to the edge of the surface not being particularly easily controlled), or
- using an ad hoc variation of the rules at the edge (which gives good control of the position of the edge but no control of the first derivative across it, and zero second derivative, leading to bad curvature plots).

Bézier end conditions would allow good control of both the edge itself and the tangents across it, and are described elegantly in terms of having a fully multiple knot at the end. They therefore fall naturally into a non-uniform context. However in this case, as in Section 2, we fail to find a suitable factorisation of the relevant subdivision matrices.

We discuss the reasons for this failure in Section 4 and describe a different approach from Schaefer and Goldman [4] which meets our original goals using a blossoming approach. In Section 5, we apply this technique to the Bézier end conditions of Section 3 to give a concrete example of a non-uniform subdivision matrix expressed as the product of narrow-band factors. Finally, we draw some conclusions in Section 6.

1.2 Notation

We will write τ_i for the ith knot of the original knot vector $\boldsymbol{\tau}$, and t_{2i} for the corresponding knot in the refined knot vector $\boldsymbol{t}$ ($\tau_i = t_{2i}$). t_i at odd values of i is the knot inserted between knots $\tau_{(i-1)/2}$ and $\tau_{(i+1)/2}$. As for all B-splines, we require $t_i \leq t_{i+1}$ for all i.

Let the subdivision matrix of order k (degree $k-1$) that transforms B-splines on $\boldsymbol{\tau}$ into B-splines on $\boldsymbol{t}$ be $\boldsymbol{S}^k$. Knot insertion is simply a change of basis [8], and if $B_{l,k,\boldsymbol{\gamma}}(x)$ is the lth B-spline basis function of order k on knot vector $\boldsymbol{\gamma}$, then $\boldsymbol{S}^k$ is the basis transformation matrix that gives the coordinates of each $B_{j,k,\boldsymbol{\tau}}(x)$ relative to the $B_{i,k,\boldsymbol{t}}(x)$:

$$B_{j,k,\boldsymbol{\tau}}(x) = \sum_i S^k_{ij} B_{i,k,\boldsymbol{t}}(x) \tag{1}$$

In application to subdivision, the coefficient that multiplies the jth control point in contribution to the ith new control point is therefore S^k_{ij}. (The original description of the Oslo algorithm [2] uses the notation $\alpha_{jk}(i)$ for S^k_{ij}.) Varying i for a given j produces a *mask*, and varying j for a given i produces a *stencil*.

2 Factorising Non-uniform Knot Insertions

Uniform subdivision generalises, in the non-uniform case, to knot insertion, first published as the 'Oslo algorithm' by Cohen et al. [2], and independently as the Boehm algorithm [1]. There has been little work so far, however, on generalising Lane and Riesenfeld's work [7] to non-uniform subdivision by varying smoother mask coefficients in a similar way. Goldman et al. [5] present a construction for knots in geometric series, but do not consider the general case. Warren [11] considers a framework for non-uniform knot sequences, but limits knot vector refinement to midpoint insertion, and Gregory et al. [6] derive results for a non-uniform 'corner cutting' procedure. Neither of these approaches is based on B-spline knot insertion, however, which is required for compatibility with NURBS. In this paper, we consider two generalisations which *are* based on knot insertion: our own non-uniform refine and smooth, and Schaefer's algorithm [4].

2.1 Oslo Knot Insertion

We can find $\boldsymbol{S}^k$ in several equivalent ways. To begin with, since $k = 1$ gives a piecewise constant B-spline, we can see that $\boldsymbol{S}^1$ for our $\boldsymbol{\tau}$ and $\boldsymbol{t}$ is the matrix which duplicates control points (the 'refine' step of Lane-Riesenfeld):

$$\begin{bmatrix} \ddots & & \\ & 1 & 0 \\ & 1 & 0 \\ & 0 & 1 \\ & 0 & 1 \\ & & \ddots \end{bmatrix} \tag{2}$$

For $k > 1$, the Oslo algorithm [2] calculates the values that can be non-zero for a row of $\boldsymbol{S}^k$ using a triangular scheme. The sum of the values in each row of $\boldsymbol{S}^k$ is always 1, and each row is therefore 'smeared out' as k increases — echoing, in a discrete way, the continuous convolution of B-spline basis functions. There is a triangle generated by the Oslo algorithm for each row of $\boldsymbol{S}^k$, with an apex at each '1' of $\boldsymbol{S}^1$.

2.2 Understanding Uniform Refine and Smooth

We want to formulate the properties of a *uniform* refine and smooth factorisation in terms of the knot insertion matrices $\boldsymbol{S}^k$, and then generalise to a non-uniform factorisation with similar properties. To do so, we need to look at the following appealing property of the Lane-Riesenfeld algorithm.

– After d smoothing steps, the points are the control polygon for a uniform B-spline of degree d which matches the B-spline of the same degree (with knot intervals doubled) defined on the original polygon.

This property cannot be generalised in terms of constant knot vectors $\boldsymbol{\tau}$ and $\boldsymbol{t}$. To see this, consider a single knot interval $[\tau_i\ \tau_{i+1}]$. The knot t_{2i+1} is inserted here and the relevant part of $\boldsymbol{S}^1$ (for *constant* basis functions) is $(\begin{smallmatrix}\frac{1}{1}\end{smallmatrix})$, duplicating the control point which corresponds to the original interval. The result of smoothing these values in any ratio is therefore to return the same original point. *Linear* basis functions, however, have a support of two knot intervals, so within $[\tau_i\ \tau_{i+1}]$ there is only one new basis function and the relevant part of $\boldsymbol{S}^2$ is $(\begin{smallmatrix}\frac{1}{2}\ \frac{1}{2}\end{smallmatrix})$, taking a mean of adjacent points.

This problem arises because where k is even, control points correspond to knot *values*. Where k is odd, they are paired with knot *intervals*. We therefore cannot hope to compare $\boldsymbol{S}^j$ with $\boldsymbol{S}^k$ in the non-uniform case if $|j - k|$ is odd. To do so, we would need to either change $\boldsymbol{\tau}$ and $\boldsymbol{t}$ or violate symmetry.

Bearing this in mind, we might list properties of a uniform refine and smooth factorisation as follows. We take $\theta = 1 + (k + 1 \bmod 2)$, so $\theta = 1$ where k is odd, and 2 where k is even.

A. For any k, there are smoothing matrices $\boldsymbol{M}^\theta, \ldots, \boldsymbol{M}^{k-1}$ such that $\boldsymbol{S}^k = \boldsymbol{M}^{k-1}\boldsymbol{M}^{k-2}\ldots\boldsymbol{M}^\theta\boldsymbol{S}^\theta$.

B. Furthermore, for each $\kappa < k$ such that $k - \kappa$ is even, $\boldsymbol{M}^{\kappa-1} \dots \boldsymbol{M}^{\theta} \boldsymbol{S}^{\theta} = \boldsymbol{S}^{\kappa}$.
C. Each $\boldsymbol{M}$ is a band matrix of bandwidth 2.
D. Each row of each $\boldsymbol{M}$ performs $\frac{1}{2}, \frac{1}{2}$ averaging.

No factorisation for non-uniform knot insertion can hold all four of these properties, so we are looking for a generalisation which maintains a chosen subset. In particular, property A makes the factorisation useful and property C is important for the benefits of locality discussed in Section 1. In Section 4, we will consider Schaefer's generalisation [4] of refine and smooth which holds just these two of the four. In this section, we consider a non-uniform analogue of the Lane-Riesenfeld approach which also maintains property B.

2.3 Our Generalised Refine and Smooth

Since our non-uniform refine and smooth will not hold property D, we allow each row $\boldsymbol{M}_i^k$ of the smoothing matrices to hold a different ratio. These ratios may differ within the same matrix and between different smoothing matrices — we only require that the weights in each row still sum to 1. Taking affine combinations of points is necessary in order to retain invariance under solid-body (and affine) transformations.

We can also consider the product of two smoothing matrices $\boldsymbol{M}^{k+1}\boldsymbol{M}^k$ as a single smoothing matrix which takes a weighted mean of three rows. If the weights in the ith row of the smoother matrix that gives $\boldsymbol{S}^{d+2}$ from $\boldsymbol{S}^d$ are α_i^d, β_i^d and $1 - (\alpha_i^d + \beta_i^d)$, then we require

$$\begin{aligned} & \alpha_i^d \boldsymbol{S}_i^d + \beta_i^d \boldsymbol{S}_{i+1}^d + (1 - \alpha_i^d - \beta_i^d)\boldsymbol{S}_{i+2}^d = \boldsymbol{S}_i^{d+2} \\ \text{i.e.} \quad & \alpha_i^d(\boldsymbol{S}_i^d - \boldsymbol{S}_{i+2}^d) + \beta_i^d(\boldsymbol{S}_{i+1}^d - \boldsymbol{S}_{i+2}^d) = \boldsymbol{S}_i^{d+2} - \boldsymbol{S}_{i+2}^d \end{aligned} \tag{3}$$

Or expressed as a system of equations (which must be satisfied for all j),

$$\begin{pmatrix} \alpha_i^d & \beta_i^d \end{pmatrix} \begin{pmatrix} \boldsymbol{S}_i^d - \boldsymbol{S}_{i+2}^d \\ \boldsymbol{S}_{i+1}^d - \boldsymbol{S}_{i+2}^d \end{pmatrix} = (\boldsymbol{S}_i^{d+2} - \boldsymbol{S}_{i+2}^d). \tag{4}$$

We only want to look at every second d. If k is even, $d \in \{2, 4, \dots, k\}$ and if k is odd, $d \in \{1, 3, \dots, k\}$. This means that the 'refine' step of our generalisation is $\boldsymbol{S}^{\theta}$ from property A. If the resulting systems of equations (4), one for each value of j, has a solution for α_i^d and β_i^d for every i and relevant d, then we will say that the knot vector refinement $\boldsymbol{t}$ of $\boldsymbol{\tau}$ has a refine and smooth formulation.

2.4 Two Stencils to Consider

In order to characterise the behaviour of our refine and smooth factorisation, we now need to establish the values of i we need to consider in the systems (4). Each system sets up a correspondence between a set of four rows: three in $\boldsymbol{S}^d$ and another in $\boldsymbol{S}^{d+2}$. There is no need to examine two such sets if there is an isomorphism between the two.

As our $\boldsymbol{\tau}$ and $\boldsymbol{t}$ perform binary subdivision, there are only two types of stencil appearing in each $\boldsymbol{S}^k$, and there are therefore two differing sets of rows to consider. At $\boldsymbol{S}^1$, the two groups of three rows, which are both smoothed to obtain a row of $\boldsymbol{S}^3$, are

$$\begin{matrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{matrix} \quad \text{and} \quad \begin{matrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{matrix}. \tag{5}$$

Every other collection of three rows is isomorphic to one of these cases, where the required mapping involves just a shift and subscript-rewriting. We can obtain the row entries for both cases (5) using the Oslo algorithm [2] and check every resulting linear system (4) to a given depth k.

The number of equations to be satisfied in a system (4) depends on the width of the stencils in the relevant rows. These stencils grow wider with k, so we expect the systems to become more constrained as k grows. If k is large enough, we may require additional constraints on $\boldsymbol{\tau}$ and $\boldsymbol{t}$ in order to satisfy the systems, as each smoother has just two degrees of freedom.

2.5 Factorisable Knot Insertions

We now have a target factorisation which expresses a knot insertion matrix $\boldsymbol{S}^k$ as the product of a refinement matrix and $\lfloor \frac{k-1}{2} \rfloor$ smoothing matrices. To establish for which knot insertion cases this factorisation exists, we used MATLAB to analyze the systems (4) in the two cases (5) and find that

- for $k \leq 5$, our refine and smooth formulation exists for every possible knot insertion.
- for $k \geq 6$ (degree at least quintic), equations limit the possible configurations for $\boldsymbol{\tau}$ and $\boldsymbol{t}$.

We can find an equation that relates knots in both $\boldsymbol{\tau}$ and $\boldsymbol{t}$, but the analysis is more manageable when considering specific configurations for $\boldsymbol{\tau}$. For the rest of this section we will be considering the case for $k = 6$, a quintic B-spline, where we take $\boldsymbol{\tau}$ to be uniform, and then a sequence in geometric series. The results for these specific $\boldsymbol{\tau}$ provide some insight into the general behaviour.

Knot insertion on a uniform knot vector. For a uniform knot vector, we set $\tau_i = i$. Instead of writing down a different equation for different positions, we will give a single equation that characterises the constraints for a whole knot vector. To do so, we write $r_i = (t_{2i+1} - \tau_i)/(\tau_{i+1} - \tau_i)$. Uniform $\boldsymbol{\tau}$ then gives $r_i = t_{2i+1} - i$, and we find that our refine and smooth formulation requires for every i

$$\begin{aligned} 0 = {} & 4r_i^3 - 3r_i^2(r_{i-1} + 1 + r_{i+1}) \\ & + 2r_i(2r_{i-1} + 1 + r_{i-1}r_{i+1}) - r_{i-1}r_{i+1} - 2r_{i-1} \end{aligned} \tag{6}$$

Figure 1 shows a contour plot of r_{i+1} in terms of r_{i-1} and r_i. For $\boldsymbol{\tau}$ and $\boldsymbol{t}$ to have the format described in Section 1.2, we require $0 \leq r_{i+1} \leq 1$. Figure 1

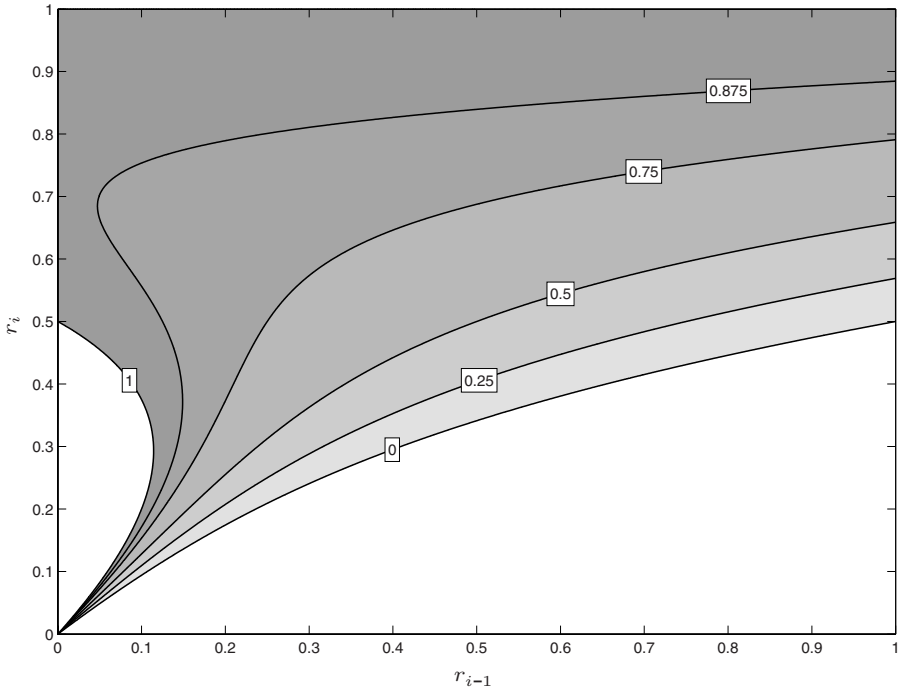


Fig. 1. The value of r_{i+1} to produce a knot insertion into a uniform knot vector with a refine and smooth formulation, when constrained by r_{i-1} and r_i. The shaded region gives $0 \leq r_{i+1} \leq 1$ and therefore shows the valid values for (r_{i-1}, r_i).

therefore shows the boundary for values of r_{i-1} and r_i which lead to viable values for r_{i+1}. For a refined knot vector with more than 7 knots[1], we must also consider the equation that constrains r_i, $r_i + 1$ and $r_i + 2$, which is identical to (6) with $i + 1$ for i. In a similar fashion, we can consider the effect of r_{i-1} and r_i on r_{i-2}. The area of possible (r_{i-1}, r_i) values shrinks, the more knots there are in $\boldsymbol{t}$.

We already know, though, that $r_i = \frac{1}{2}$ for all i is a refine and smooth-factorisable knot insertion, irrespective of the length of $\boldsymbol{t}$. This is uniform knot insertion into a uniform knot vector — exactly Lane-Riesenfeld subdivision [7]. In fact, it is simple to verify from (6) that $r_i = v$ for all i is also a refine and smooth knot insertion that preserves property B, for any $0 \leq v \leq 1$. We conjecture that this will be true for a uniform knot vector at any degree.

Knot insertion on a knot vector in geometric series. We now turn to a knot vector in geometric series with ratio 2, by setting $\tau_i = 2^i$. We can still completely characterise the constraints on $\boldsymbol{t}$ using an equation in terms of the

[1] 3 inserted knots and the 4 original uniform knots.

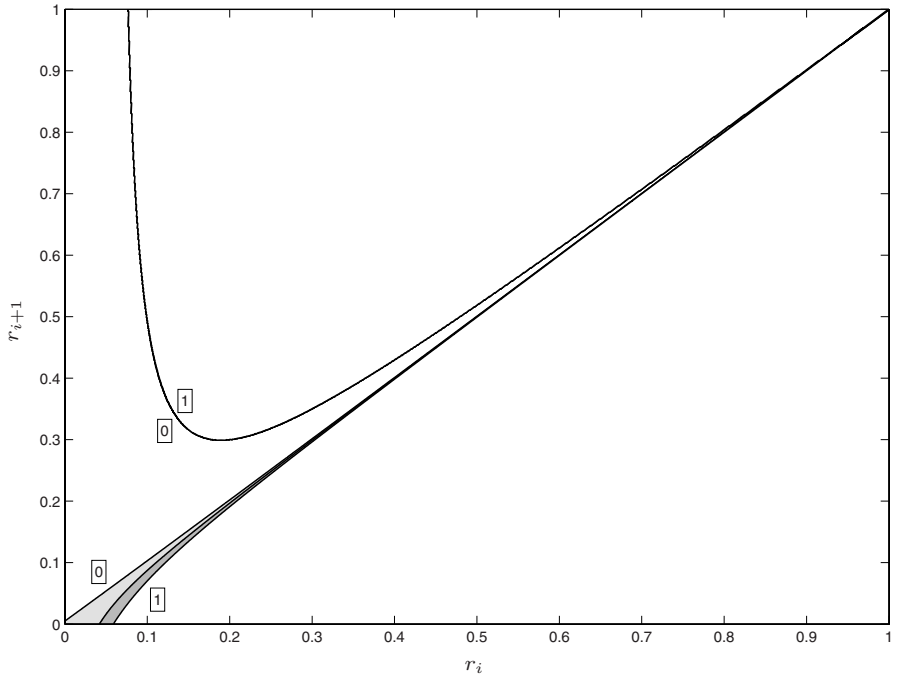


Fig. 2. The value of r_{i-1} to produce a knot insertion into a geometric series knot vector with a refine and smooth formulation, when constrained by r_i and r_{i+1}

ratios r_i (now $r_i = (t_{2i+1}/2^i) - 1$), but the equation to be satisfied is not as neat as (6):

$$\begin{aligned} 0 = {} & 4369r_i^3 - 3r_i^2(91r_{i-1} - 334 + 1456r_{i+1}) \\ & + r_i(22r_{i-1} + 5 - 1008r_{i+1} + 272r_{i-1}r_{i+1}) - 16r_{i-1}r_{i+1} - 5r_{i-1} \end{aligned} \tag{7}$$

Figure 2 shows the contour plot of r_{i-1} in terms of r_i and r_{i+1}, as this puts far greater constraints on the possible values of (r_i, r_{i+1}) than the value of r_{i+2} does. We can see that in the geometric case, the two degrees of freedom in assigning (r_i, r_{i+1}) are nearly reduced to just one, as little deviation is possible away from the solution $r_i = r_{i+1}$. However, in a surprising similarity with the uniform case, we can still show that $r_i = v$ for all i is a refine and smooth knot insertion, for any $0 \leq v \leq 1$, and regardless of the length of $\boldsymbol{t}$.

This is immediately useful when inserting into a geometric series knot vector, as we can choose v such that the refined knot vector is also in geometric series (here, we would have $v = \sqrt{2} - 1$). The refined knot vector can then be mapped onto the original using just scales and shifts in parameter, and so it is possible to subdivide the *refined* knot vector (using refine and smooth) with any $0 \leq v \leq 1$ as well. This was first shown by Goldman and Warren [5].

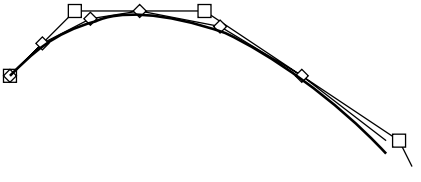
3 Bézier End Conditions

These constraints on factorisable knot insertions make our refine and smooth formulation unusable in practice. We consider the general problem again in Section 4, but first we look at the same approach in the light of a different context; that of Bézier end conditions.

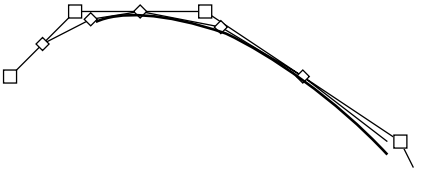
If the first and last knots in an otherwise uniform knot-vector have multiplicity of at least the degree of the B-spline, the ends of the curve behave in a very similar way to those of a Bézier curve. The end control point determines the position of the end of the curve, the end two determine the first derivative, the end three the second derivative and so on. In fact if there are no internal knots, the curve is a Bézier curve. Because this scenario is just a particular case of the general non-uniform B-spline, all the properties of affine invariance and containment in the convex hull are retained.

This nice behaviour should be contrasted with the ways in which subdivision curves are usually terminated;

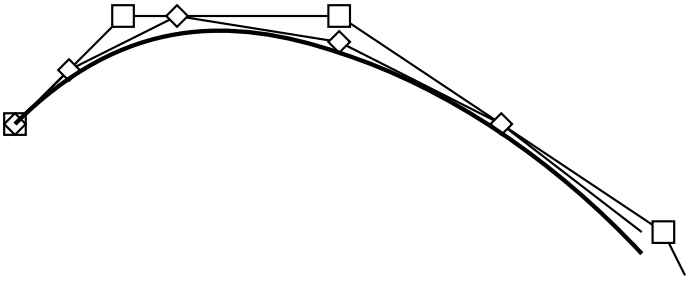
– truncation, where the only control points after one refinement are those which are well-defined by the same rules as are used in the interior of the curve. The polygon gets shorter at every refinement, and the end of the limit curve can only be precisely positioned by solving a linear system to determine where the outer control points should be.



(a) Ad hoc end with zero curvature



(b) Truncated polygon



(c) Bézier end-condition

Fig. 3. Cubic limit curves with their control polygons

– The usual ad hoc fixup, where control points for which the general rules cannot be applied because their antecedents do not exist, are given special rules. In particular in the cubic case, only the end control point after refinement is undefined, and it is typically positioned at the end point before refinement. This enables easy positioning of the end of the limit curve, but results in the second derivative being zero there.

We would like to have a systematic approach which can be used to either build explicit subdivision matrices for Bézier end conditions, or, hopefully, to be just a particular case within the refine and smooth paradigm.

3.1 Systematic Development

We are looking at setting, for example, $\tau_i = 0$ for $i \leq 0$ and $\tau_i = 2i, t_i = i$ for $i \geq 1$. Examples of the resulting $\boldsymbol{S}^k$ are given for $k = 2 \ldots 5$ in Fig. 4, and a method for computation of these specific $\boldsymbol{S}^k$ is described in [3].

Linear, $k = 2$

$$\frac{1}{2}\begin{bmatrix} 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & 1 & \\ & & & \ddots \end{bmatrix}$$

Quadratic, $k = 3$

$$\frac{1}{4}\begin{bmatrix} 4 & & & & \\ 2 & 2 & & & \\ & 3 & 1 & & \\ & 1 & 3 & & \\ & & 3 & 1 & \\ & & 1 & 3 & \\ & & & & \ddots \end{bmatrix}$$

Cubic, $k = 4$

$$\frac{1}{16}\begin{bmatrix} 16 & & & & & \\ 8 & 8 & & & & \\ & 12 & 4 & & & \\ & 3 & 11 & 2 & & \\ & & 8 & 8 & & \\ & & 2 & 12 & 2 & \\ & & & & & \ddots \end{bmatrix}$$

Quartic, $k = 5$

$$\frac{1}{48}\begin{bmatrix} 48 & & & & & & \\ 24 & 24 & & & & & \\ & 36 & 12 & & & & \\ & 9 & 33 & 6 & & & \\ & & 20 & 25 & 3 & & \\ & & 4 & 29 & 15 & & \\ & & & 15 & 30 & 3 & \\ & & & 3 & 30 & 15 & \\ & & & & & & \ddots \end{bmatrix}$$

Fig. 4. The knot insertion matrices $\boldsymbol{S}^k$ for $k = 2 \ldots 5$ and Bézier end conditions

In the uniform case (with cyclic data from a closed polygon) we can determine $\boldsymbol{S}^{k+1}$ from $\boldsymbol{S}^k$ merely by multiplying by the circulant smoothing matrix

$$\begin{bmatrix} \ddots & & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & & \ddots \end{bmatrix}$$

We are interested in seeing whether we can derive $\boldsymbol{S}^{k+1}$ from $\boldsymbol{S}^k$ in a similar manner in the Bézier end condition case. For the reasons discussed in Section 1, we would still like the matrix that multiplies $\boldsymbol{S}^k$ to have a narrow band. In order to derive this multiplier, we observe three interesting properties from the examples in Fig. 4:

1. that the top left $k \times k$ square of $\boldsymbol{S}^k$ is identical to the top left corner of $\boldsymbol{S}^{k+1}$ (remember the global scaling factor on each matrix).
2. that beyond the $(k-1)$th column the matrix is regular.
3. that beyond the $(2k-4)$th row the matrix is regular.

The interesting questions are

– whether these matrices can be determined systematically other than by carrying out the full knot insertion process for each degree.
– whether the refine and smooth paradigm, which is so elegant for the implementation of high degree uniform B-splines as subdivision curves, can be extended to cover this case.

Now if $\boldsymbol{S}^{k+1} = \boldsymbol{M}\boldsymbol{S}^k$ we would expect to be able to derive $\boldsymbol{M}$ from the product $\boldsymbol{S}^{k+1}(\boldsymbol{S}^k)^{-1}$, but this is not straightforward. Each of the $\boldsymbol{S}^k$s is about twice as high as it is wide, and therefore inversion is not a well defined process. In fact $\boldsymbol{M}$ has about twice the number of elements as the number of conditions we are trying to satisfy, and so it is heavily underdetermined.

We are actually looking for a sparse structure, and just using, for example, the Penrose pseudo-inverse does not give the sparsity that would make the factorisation useful.

Property 1, above, suggests that the top left corner of $\boldsymbol{M}$ should be a unit matrix, and that the remainder of the first $k-1$ rows should be zero. Properties 2 and 3 suggest that beyond the $(2k-4)$th row, the pattern of uniform smoothing should be present. The question is what happens in between. We look at a couple of example cases.

Quadratic to Cubic

$$\left[\begin{array}{cccc|c} 16 & & & & \\ 8 & 8 & & & \\ & 12 & 4 & & \\ \hline & 3 & 11 & 2 & \\ & & 8 & 8 & \\ & & 2 & 12 & 2 \\ & & & & \ddots \end{array}\right] = \left[\begin{array}{ccccc|ccc|c} 4 & & & & & & & & \\ & 4 & & & & & & & \\ & & 4 & & & & & & \\ \hline -1 & 2 & -1 & & & 2 & 2 & & \\ & & & & & & 2 & 2 & \\ & & & & & & & 2 & \\ & & & & & & & & \ddots \end{array}\right] \left[\begin{array}{ccc|ccc|c} 4 & & & & & & \\ 2 & 2 & & & & & \\ & 3 & 1 & & & & \\ \hline & 1 & 3 & & & & \\ & & 3 & 1 & & & \\ & & 1 & 3 & & & \\ & & & & & & \ddots \end{array}\right]$$

Cubic to Quartic

$$\left[\begin{array}{cccc|ccccc} 48 & & & & & & & & \\ 24 & 24 & & & & & & & \\ & 36 & 12 & & & & & & \\ & 9 & 33 & 6 & & & & & \\ \hline & & 20 & 25 & 3 & & & & \\ & & 4 & 29 & 15 & & & & \\ & & & 15 & 30 & 3 & & & \\ & & & 3 & 30 & 15 & & & \\ & & & & & & \ddots & & \end{array}\right]$$

$$= \frac{1}{8} \left[\begin{array}{cccc|ccccc} 24 & & & & & & & & \\ & 24 & & & & & & & \\ & & 24 & & & & & & \\ & & & 24 & & & & & \\ \hline 45 & -90 & 65 & -20 & 12 & 12 & & & \\ 9 & -18 & 13 & -4 & & 12 & 12 & & \\ & & & & & & 12 & 12 & \\ & & & & & & & 12 & \\ & & & & & & & & \ddots \end{array}\right] \left[\begin{array}{ccccc|ccc} 16 & & & & & & & \\ 8 & 8 & & & & & & \\ & 12 & 4 & & & & & \\ & 3 & 11 & 2 & & & & \\ \hline & & 8 & 8 & & & & \\ & & 2 & 12 & 2 & & & \\ & & & 8 & 8 & & & \\ & & & 2 & 12 & 2 & & \\ & & & & & & \ddots & \end{array}\right]$$

3.2 The Need for a Different Approach

For Bézier end conditions, the matrix $\boldsymbol{M}$ by which we multiply the order k subdivision matrix to get the order $k+1$ subdivision matrix does not appear to have a derivable narrow band in the region where the two kinds of regularity meet. These $\boldsymbol{M}$ matrices are therefore performing non-local operations, and so are not useful in the way described in Section 1.

For the property-B-preserving general case explored in Section 2, the result is even worse. Although we can enforce local smoothing operations (property C), we must pay for the privilege on two counts. Firstly, we create **knot dependencies**; in the quintic case, positioning two inserted knots determines every other inserted knot. Secondly, the resulting structure is **restricted**; not all knot insertions are factorisable in this way.

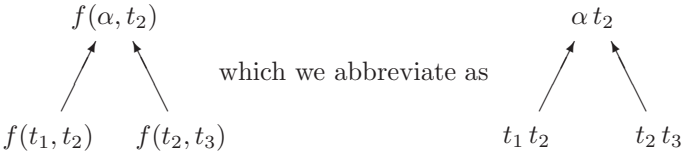
We want the position of inserted knots to be a design decision, influenced by factors such as convergence to uniformity. Instead we have a system where the knot insertion machinery prescribes the position of inserted knots, and this is not a viable route for further investigation.

4 Schaefer's Knot Insertion Algorithm

The work of Goldman and Schaefer [4] throws new light on these results by showing the impact of retaining property B described in Section 2. In this section, we describe *Schaefer's knot insertion* algorithm, which constructs an analogue of the Lane-Riesenfeld algorithm for arbitrary knot vectors without suffering from the restrictions, loss of locality or knot dependencies seen in Sections 2 and 3.

The algorithm achieves this by seeking to maintain only properties A and C of uniform refine and smooth in the non-uniform context.

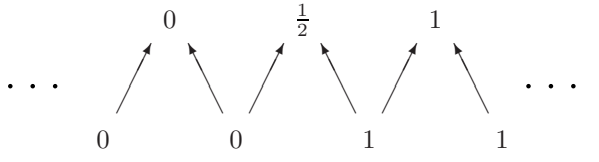
Schaefer's algorithm has a direct blossoming [9] proof and derivation. It is best illustrated using diagrams of the following format (this example is for a quadratic spline):



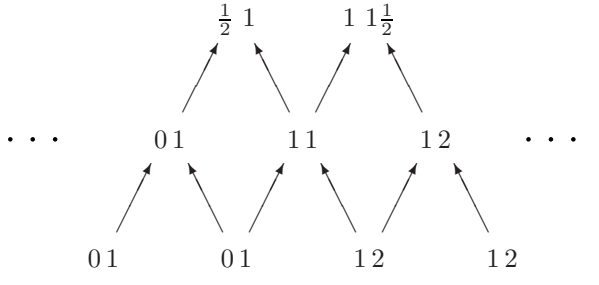
This diagram represents the affine combination of blossoms

$$f(\alpha, t_2) = \frac{t_3 - \alpha}{t_3 - t_1} f(t_1, t_2) + \frac{\alpha - t_1}{t_3 - t_1} f(t_2, t_3) \tag{8}$$

We recall that B-spline control points are given by blossoms of the form $f(\tau_i, \ldots, \tau_{i+k-2})$ for adjacent knots. Within this scheme, linear Lane-Riesenfeld subdivision is represented by the diagram



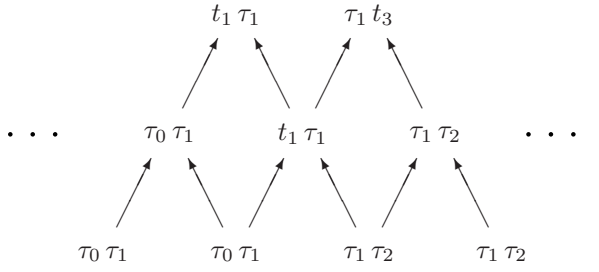
Blossoms are repeated at the bottom level of the diagram, as duplicating control points is the first step in Lane-Riesenfeld refine and smooth. The following diagram subsequently employs two smoothing steps, for quadratic Lane-Riesenfeld subdivision:



At the top level of the diagram, blossoms are produced with adjacent knots in a *refined* knot vector, with half the spacing of the original knots.

In both the linear and quadratic cases, Lane-Riesenfeld subdivision can be extended to non-uniform knot vectors simply by substituting the non-uniform values into the blossom diagrams above. For degree greater than quadratic, however, it becomes hard to prove properties of Lane-Riesenfeld refine and smooth with these diagrams. Schaefer's knot insertion, by contrast, can be readily proved

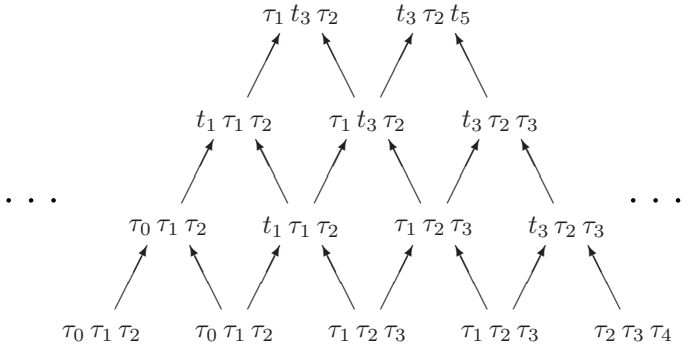
with blossoming diagrams for arbitrary degree. This is the diagram which represents Schaefer's knot insertion for quadratics:



There are several important properties of Schaefer's knot insertion factorisation:

- Knots are inserted "as soon as possible". The Lane-Riesenfeld algorithm calculates all the required points in the final smoothing step, whereas Schaefer's algorithm calculates half the points in the penultimate smoothing, and the remaining half in the final step.
- Half of the affine combinations in any given row simply select one of the blossoms from the row below. In this sense, Schaefer's knot insertion is faster than Lane-Riesenfeld, as it only performs half the work.
- Schaefer's knot insertion does not specialise to the Lane-Riesenfeld construction in the case where knot spacings are uniform. Goldman [4] presents two algorithms for cubic knot insertion which do have this property, but have not been extended to general degree.
- The central blossom value in the above diagram could equally have been $\tau_1\,t_3$, and Schaefer's algorithm makes an arbitrary choice between these two asymmetric options.

A final property of Schaefer's algorithm, which is attractive for the sake of understanding, is that the diagram representing the cubic algorithm is readily derived from the quadratic diagram by simply appending blossom arguments. To make this clear, here is the diagram for the cubic case:



5 A Factorisation for Bézier End Conditions

We can now apply Schaefer's algorithm from Section 4 to the Bézier end conditions considered in Section 3. This illustrates that Schaefer's algorithm provides a framework for implementing subdivision curves and surfaces, with Bézier end conditions, using a refine and smooth method. Here we provide the factorisations for linear, quadratic and cubic curves. Since Schaefer's algorithm does not reduce to Lane-Riesenfeld for uniform knots, the smoothing matrices do not compute arithmetic means in the regular regions.

Linear

$$\frac{1}{2}\begin{bmatrix} 2 & \\ 1 & 1 \\ & 2 \\ & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & 2 & & \\ & 1 & 1 & \\ & & 0 & 2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \end{bmatrix}$$

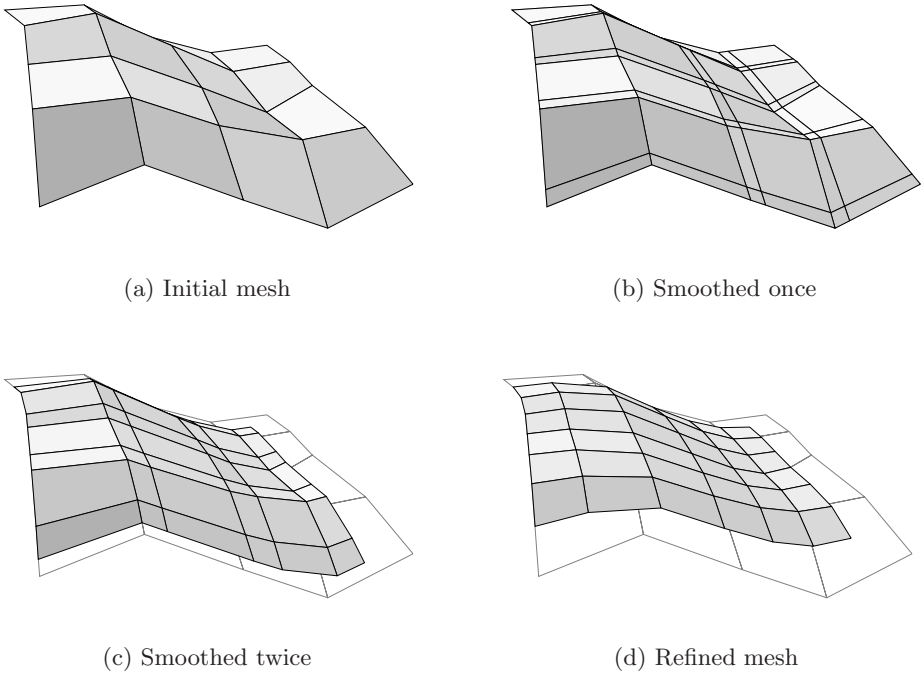
Quadratic

$$\frac{1}{4}\begin{bmatrix} 4 & & \\ 2 & 2 & \\ & 3 & 1 \\ & 1 & 3 \\ & & 3 \\ & & 1 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 0 & 6 & & & & \\ & 3 & 3 & & & \\ & & 0 & 6 & & \\ & & & 2 & 4 & \\ & & & & 0 & 6 \\ & & & & & 2 \end{bmatrix} \frac{1}{4}\begin{bmatrix} 0 & 4 & & & & \\ & 4 & 0 & & & \\ & & 0 & 4 & & \\ & & & 3 & 1 & \\ & & & & 0 & 4 \\ & & & & & 3 \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} 1 & & & \\ 1 & & & \\ & 1 & & \\ & 1 & & \\ & & 1 & \\ & & 1 & \\ & & & 1 \\ & & & 1 \end{bmatrix}$$

Cubic

$$\frac{1}{16}\begin{bmatrix} 16 & & & \\ 8 & 8 & & \\ & 12 & 4 & \\ & 3 & 11 & 2 \\ & & 8 & 8 \\ & & 2 & 12 \\ & & & 2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 0 & 4 & & & & & \\ & 2 & 2 & & & & \\ & & 0 & 4 & & & \\ & & & 1 & 3 & & \\ & & & & 0 & 4 & \\ & & & & & 1 & 3 \end{bmatrix} \frac{1}{20}\begin{bmatrix} 0 & 20 & & & & & & \\ & 20 & 0 & & & & & \\ & & 0 & 20 & & & & \\ & & & 15 & 5 & & & \\ & & & & 0 & 20 & & \\ & & & & & 12 & 8 & \\ & & & & & & 0 & 20 \end{bmatrix}$$

$$\times \frac{1}{6}\begin{bmatrix} 0 & 6 & & & & & & \\ & 6 & 0 & & & & & \\ & & 0 & 6 & & & & \\ & & & 6 & 0 & & & \\ & & & & 0 & 6 & & \\ & & & & & 5 & 1 & \\ & & & & & & 0 & 6 \\ & & & & & & & 5 \\ & & & & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ 1 & & & & \\ & 1 & & & \\ & 1 & & & \\ & & 1 & & \\ & & 1 & & \\ & & & 1 & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$



(a) Initial mesh (b) Smoothed once

(c) Smoothed twice (d) Refined mesh

Fig. 5. A non-uniform bicubic subdivision step using Schaefer's algorithm. The left-hand edge has a Bézier end condition.

Figure 5 shows this factorisation in one direction of a tensor-product application of Schaefer's algorithm. The orthogonal direction uses the factorisation for uniform subdivision.

6 Conclusions

- A refine and smooth factorisation of knot insertion matrices is useful for implementation, and may prove a useful step towards non-uniform subdivision schemes of general degree.
- The definition of "refine and smooth factorisation" can make a marked difference to what is possible and what the factorisation looks like.
- Schaefer's algorithm performs non-uniform knot insertion for any degree, using a refine and smooth factorisation, and is easily derived using blossoming. The factorisation may prove useful conceptually as well as in implementation, and merits further investigation.

References

1. Boehm, W.: Inserting new knots into B-spline curves. Computer-Aided Design 12(4), 199–201 (1980)
2. Cohen, E., Lyche, T., Riesenfeld, R.: Discrete *B*-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics. Computer Graphics and Image Processing 14(2), 87–111 (1980)
3. Gasciola, G., Romani, L.: A general matrix representation for non-uniform B-spline subdivision with boundary control. Draft paper (2006)
4. Goldman, R., Schaefer, S.: Global Knot Insertion Algorithms. Presentation at the University of Kaiserslautern (January 17, 2007)
5. Goldman, R., Warren, J.: An extension of Chaiken's algorithm to B-spline curves with knots in geometric progression. CVGIP: Graphical Models and Image Processing 55(1), 58–62 (1993)
6. Gregory, J., Qu, R.: Nonuniform corner cutting. Computer Aided Geometric Design 13(8), 763–772 (1996)
7. Lane, J., Riesenfeld, R.: A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 2(1), 35–46 (1980)
8. Lyche, T., Morken, K.: Making the OSLO Algorithm More Efficient. SIAM Journal on Numerical Analysis 23(3), 663–675 (1986)
9. Ramshaw, L.: Blossoming: A Connect-the-Dots Approach to Splines. Technical Report 19, Digital Systems Research Center (1987)
10. Sederberg, T., Zheng, J., Sewell, D., Sabin, M.: Non-Uniform Recursive Subdivision Surfaces. In: Proceedings of the 25th annual conference on Computer Graphics and Interactive Techniques, pp. 387–394 (1998)
11. Warren, J.: Binary subdivision schemes for functions over irregular knot sequences. In: Dæhlen, M., Lyche, T., Schumaker, L.L. (eds.) Mathematical Methods for Curves and Surfaces. Vanderbilt U.P, 543–562 (1995)

Scattered Data Fitting on Surfaces Using Projected Powell-Sabin Splines

Oleg Davydov[1] and Larry L. Schumaker[2]

[1] Department of Mathematics, University of Strathclyde,
26 Richmond Street, Glasgow, G1 1XH, UK
oleg.davydov@strath.ac.uk
[2] Department of Mathematics, Vanderbilt University, Nashville, TN 37240, USA
larry.schumaker@vanderbilt.edu

Abstract. We present C^1 methods for either interpolating data or for fitting scattered data associated with a smooth function on a two-dimensional smooth manifold Ω embedded into $\mathbb{R}^3$. The methods are based on a local bivariate Powell-Sabin interpolation scheme, and make use of local projections on the tangent planes. The data fitting method is a two-stage method. We illustrate the performance of the algorithms with some numerical examples, which, in particular, confirm the $\mathcal{O}(h^3)$ order of convergence as the data becomes dense.

1 Introduction

Let Ω be a 2-dimensional smooth manifold. For simplicity we assume that Ω is compact and has no boundary. Suppose we are given the values of a (possibly unknown) smooth function f defined on Ω at a set of points X on Ω. Our aim is to construct a function s defined on Ω that approximates f. This problem arises frequently in practice, see Remark 1, but there do not seem to be many methods available for general manifolds. Several methods have been developed for the case when Ω is the sphere, see Remark 2.

Our approach to solving this problem is as follows. Suppose we have an atlas $\Phi = \{(U_\xi, \phi_\xi)\}_{\xi\in\Omega}$ for Ω, where for each $\xi \in \Omega$, U_ξ are open sets on Ω containing ξ, and ϕ_ξ are mappings of U_ξ into $\mathbb{R}^2$. We assume that the ϕ_ξ depend smoothly on ξ in a certain sense, see [7]. Then for each $\xi \in \Omega$, we project the data locations into $\phi_\xi(U_\xi) \subset \mathbb{R}^2$, and use a local bivariate Powell-Sabin spline to compute the value $s(\xi)$ of the approximating function s. The resulting function s is C^1 on Ω, and it has the usual approximation properties of the Powell-Sabin spline. This approach is related to methods for interpolation and data fitting on manifolds introduced by Demjanovich [10,11] and Pottmann [25], see Remarks 3 and 4.

In this paper we examine the case when Ω is a C^2-surface, i.e. a compact 2-dimensional C^2-manifold embedded in $\mathbb{R}^3$. Our implementation is based on a natural atlas Φ defined by local projections onto the tangent planes at all points on the surface Ω, see Section 2.1. General smooth 2-manifolds are treated in [7], where we also prove the main theoretical results about smoothness of s and error bounds for interpolation and data fitting.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 138–153, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The paper is organized as follows. In Section 2 we introduce some basic concepts and notation. In Section 3 we first present a method for constructing an interpolant to the data on any smooth 2-dimensional manifold Ω embedded in $\mathbb{R}^3$, assuming we are also given values for the gradients at each of the data points in X. Next, we describe a two-stage data fitting method that works only with the data $\{f_\xi\}_{\xi\in X}$, and which is more appropriate than interpolation for large data sets or noisy data. Theoretical results on C^1 smoothness of the interpolating or approximating functions on Ω, as well as error bounds for the approximation of smooth functions are given here without proof. Numerical examples are presented in Section 4 for both the sphere and for certain ring-type manifolds. We conclude the paper with remarks and references.

2 Preliminaries

2.1 Projection Atlas

Let $\langle\cdot,\cdot\rangle$ be the usual inner product in $\mathbb{R}^3$, and let $\|a\|_2$ be the Euclidean norm of any 3-vector a. Since Ω is embedded in $\mathbb{R}^3$, it can be represented locally as a regular level surface of a C^2 function of three variables. More precisely, each point $\xi \in \Omega$ has a neighborhood $\mathcal{G}_\xi$ in $\mathbb{R}^3$ such that $\mathcal{G}_\xi \cap \Omega = F_\xi^{-1}(0)$, where $F_\xi : \mathcal{G}_\xi \to \mathbb{R}$ is a C^2 function with nonzero gradient ∇F_ξ everywhere in $\mathcal{G}_\xi \cap \Omega$, see [17]. Then $n_\xi := \nabla F_\xi(\xi)/\|\nabla F_\xi(\xi)\|_2$ is a *normal vector* to Ω at ξ. Moreover, the *tangent plane* Γ_ξ is the unique plane in $\mathbb{R}^3$ that contains ξ and is orthogonal to n_ξ. Clearly, for all $\zeta \in \mathcal{G}_\xi \cap \Omega$, a normal vector to Ω at ζ can also be computed as $\nabla F_\xi(\zeta)/\|\nabla F_\xi(\zeta)\|_2$. It coincides with either n_ζ or $-n_\zeta$. Clearly, $\langle n_\xi, \nabla F_\xi(\zeta)\rangle > 0$ for all $\zeta \in \mathcal{G}_\xi \cap \Omega$.

We are now ready to define an atlas associated with Ω. For each $\xi \in \Omega$, let U_ξ be the connected component of the open set $\{\zeta \in \Omega : \langle n_\xi, n_\zeta\rangle \neq 0\}$ that contains ξ. Then U_ξ is an open neighborhood of ξ. Clearly, the *orthogonal projection* $\pi_\xi : U_\xi \to \Gamma_\xi$ defined by

$$\pi_\xi(\zeta) = \zeta + \langle \xi - \zeta, n_\xi\rangle\, n_\xi, \qquad \zeta \in U_\xi,$$

is invertible. Assuming that $\gamma_\xi^{[1]}, \gamma_\xi^{[2]}$ are orthogonal unit vectors in Γ_ξ such that $\gamma_\xi^{[1]} \times \gamma_\xi^{[2]} = n_\xi$, we can also write

$$\pi_\xi(\zeta) = \xi + \langle \zeta - \xi, \gamma_\xi^{[1]}\rangle\, \gamma_\xi^{[1]} + \langle \zeta - \xi, \gamma_\xi^{[2]}\rangle\, \gamma_\xi^{[2]}.$$

Define ϕ_ξ by the formula

$$\phi_\xi(\zeta) := [\langle \zeta - \xi, \gamma_\xi^{[1]}\rangle, \langle \zeta - \xi, \gamma_\xi^{[2]}\rangle]^T, \qquad \zeta \in U_\xi.$$

We call $\Phi = \{(U_\xi, \phi_\xi)\}_{\xi\in\Omega}$ the *projection atlas* associated with Ω, and (U_ξ, ϕ_ξ), $\xi \in \Omega$, are its *charts*.

Let us show that Φ is indeed an atlas in the sense of the standard definition of a C^1-manifold, see e.g. [17]. This requires that $\phi_\xi : U_\xi \to \mathbb{R}^2$ is a homeomorphism

between U_ξ and an open subset of $\mathbb{R}^2$. Moreover, for every $\xi, \zeta \in \Omega$, $\phi_{\zeta\xi} := \phi_\zeta \circ \phi_\xi^{-1} : \phi_\xi(U_\zeta \cap U_\xi) \to \phi_\zeta(U_\zeta \cap U_\xi)$ should be a C^1 mapping whenever $U_\xi \cap U_\zeta \neq \emptyset$.

Clearly, ϕ_ξ is invertible by the choice of U_ξ. Consider the coordinate system for $\mathbb{R}^3$ with coordinate vectors $\gamma_\xi^{[1]}, \gamma_\xi^{[2]}, n_\xi$ and origin ξ. For any $\mu \in U_\xi$, the equation $F_\mu = 0$ determines an implicit function $x^{[3]} = \delta_\mu(x^{[1]}, x^{[2]})$ in a neighborhood of $\phi_\xi(\mu)$, such that

$$\phi_\xi^{-1}(x^{[1]}, x^{[2]}) = \xi + x^{[1]}\gamma_\xi^{[1]} + x^{[2]}\gamma_\xi^{[2]} + \delta_\mu(x^{[1]}, x^{[2]})\, n_\xi.$$

Since $\langle n_\xi, \nabla F_\mu(\mu)\rangle = \langle n_\xi, n_\mu\rangle \|\nabla F_\mu(\mu)\|_2 \neq 0$, the implicit function theorem implies that $\delta_\mu(x^{[1]}, x^{[2]})$ is a C^2 function in a neighborhood of $\phi_\xi(\mu)$. Assuming $\mu \in U_\xi \cap U_\zeta$, we also have

$$(\phi_\zeta \circ \phi_\xi^{-1})(x^{[1]}, x^{[2]}) = \phi_{\zeta\xi}(x^{[1]}, x^{[2]}) = [\phi_{\zeta\xi}^{[1]}(x^{[1]}, x^{[2]}), \phi_{\zeta\xi}^{[2]}(x^{[1]}, x^{[2]})]^T,$$

where for $i = 1, 2$,

$$\phi_{\zeta\xi}^{[i]}(x^{[1]}, x^{[2]}) = \langle \xi - \zeta + x^{[1]}\gamma_\xi^{[1]} + x^{[2]}\gamma_\xi^{[2]} + \delta_\mu(x^{[1]}, x^{[2]})\, n_\xi, \gamma_\zeta^{[i]}\rangle \qquad (1)$$

in a neighborhood of $\phi_\xi(\mu)$. Therefore $\phi_{\zeta\xi} : \phi_\xi(U_\xi \cap U_\zeta) \to \phi_\zeta(U_\xi \cap U_\zeta)$ is a C^2 mapping. Note that even if a C^1 mapping $\phi_{\zeta\xi}$ would suffice for our goals, we do need the assumption that Ω is a C^2 surface to ensure that the normal n_ξ changes smoothly with ξ, which guarantees the C^1 smoothness of our approximants, see [7].

For a C^1 function f defined in a neighborhood U of $\zeta \in \Omega$, we define $J_\zeta(f) : U \cap U_\zeta \to \mathbb{R}^{2\times 2}$ by

$$J_\zeta(f)(\mu) := J(f \circ \phi_\zeta^{-1})(\phi_\zeta(\mu)), \qquad \mu \in U \cap U_\zeta,$$

where for any smooth function $g : \mathbb{R}^2 \to \mathbb{R}^2$, $g = [g^{[1]}, g^{[2]}]^T$, $J(g)$ denotes its *Jacobian*

$$J(g) := \begin{bmatrix} \frac{\partial g^{[1]}}{\partial x^{[1]}} & \frac{\partial g^{[1]}}{\partial x^{[2]}} \\ \frac{\partial g^{[2]}}{\partial x^{[1]}} & \frac{\partial g^{[2]}}{\partial x^{[2]}} \end{bmatrix}.$$

We write

$$J_{\zeta\xi} := J_\xi(\phi_\zeta), \quad \text{on } U_\zeta \cap U_\xi,$$

so that

$$J_{\zeta\xi}(\mu) = J_\xi(\phi_\zeta)(\mu) = J(\phi_{\zeta\xi})(\phi_\xi(\mu)), \qquad \mu \in U_\zeta \cap U_\xi,$$

is the Jacobian of $\phi_{\zeta\xi}$ evaluated at $\phi_\xi(\mu)$. Since $\phi_{\zeta\xi}^{-1} = \phi_{\xi\zeta}$, the well-known properties of the Jacobian imply

$$[J_{\zeta\xi}(\mu)]^{-1} = J_{\xi\zeta}(\mu). \qquad (2)$$

For later use, we now obtain explicit formulas for the Jacobian $J_{\zeta\xi}(\xi) := J(\phi_{\zeta\xi})(\phi_\xi(\xi))$ and its determinant in the case of the projection atlas. By the

above construction, the implicit function $x^{[3]} = \delta_\xi(x^{[1]}, x^{[2]})$ is C^2 in a neighborhood of the origin $\phi_\xi(\xi)$. Moreover, it vanishes together with its gradient at the origin. Hence, by (1),

$$J_{\zeta\xi}(\xi) = [\langle \gamma_\zeta^{[i]}, \gamma_\xi^{[j]} \rangle]_{i,j=1,2}. \tag{3}$$

Clearly, the determinant of this matrix is the projection of $n_\zeta = \gamma_\zeta^{[1]} \times \gamma_\zeta^{[2]}$ on n_ξ, i.e.,

$$\det J_{\zeta\xi}(\xi) = \langle n_\zeta, n_\xi \rangle. \tag{4}$$

2.2 Projected Gradients

Let $f \in C^1(\Omega)$, and let $f_\xi = f \circ \phi_\xi^{-1}$. Since Ω is embedded in $\mathbb{R}^3$, the gradient $\nabla f_\xi = \left[\frac{\partial f_\xi}{\partial x^{[1]}}, \frac{\partial f_\xi}{\partial x^{[2]}}\right]$ of f_ξ can be identified with the 3-vector

$$\operatorname{grad} f_\xi = \frac{\partial f_\xi}{\partial x^{[1]}} \gamma_\xi^{[1]} + \frac{\partial f_\xi}{\partial x^{[2]}} \gamma_\xi^{[2]}$$

lying in the tangent plane $\Gamma_\xi \subset \mathbb{R}^3$. We write

$$\operatorname{grad}_\xi f(\mu) := (\operatorname{grad} f_\xi)(\phi_\xi(\mu)), \qquad \mu \in U_\xi,$$

for the gradient of f_ξ evaluated at $\phi_\xi(\mu)$. We call $\operatorname{grad}_\xi f(\mu)$ the *projected gradient* of f at μ. It is easy to see that $\operatorname{grad}_\xi f(\xi)$ coincides with the standard gradient of a function on a 2-surface in $\mathbb{R}^3$, as defined for example in [28, p. 96]. We also need projected gradients when $\mu \neq \xi$.

Lemma 1. *For any $\xi \in \Omega$ and $\zeta \in U_\xi$, the projected gradient $\operatorname{grad}_\zeta f(\zeta)$ is the orthogonal projection of $\operatorname{grad}_\xi f(\zeta)$ onto Γ_ζ. In particular,*

$$\operatorname{grad}_\zeta f(\zeta) = \operatorname{grad}_\xi f(\zeta) - \langle \operatorname{grad}_\xi f(\zeta), n_\zeta \rangle \, n_\zeta, \tag{5}$$

and

$$\operatorname{grad}_\xi f(\zeta) = \operatorname{grad}_\zeta f(\zeta) - \frac{\langle \operatorname{grad}_\zeta f(\zeta), n_\xi \rangle}{\langle n_\zeta, n_\xi \rangle} \, n_\zeta, \qquad \text{if } \langle n_\zeta, n_\xi \rangle \neq 0, \tag{6}$$

where n_ζ and n_ξ are the unit normal vectors to Γ_ζ and Γ_ξ, respectively.

Proof. We have

$$\operatorname{grad}_\xi f(\zeta) = \frac{\partial f_\xi}{\partial x^{[1]}}\Big(\phi_\xi(\zeta)\Big)\gamma_\xi^{[1]} + \frac{\partial f_\xi}{\partial x^{[2]}}\Big(\phi_\xi(\zeta)\Big)\gamma_\xi^{[2]}.$$

Its projection onto Γ_ζ is therefore

$$\begin{aligned}
&\left(\frac{\partial f_\xi}{\partial x^{[1]}}\Big(\phi_\xi(\zeta)\Big)\langle \gamma_\xi^{[1]}, \gamma_\zeta^{[1]} \rangle + \frac{\partial f_\xi}{\partial x^{[2]}}\Big(\phi_\xi(\zeta)\Big)\langle \gamma_\xi^{[2]}, \gamma_\zeta^{[1]} \rangle\right)\gamma_\zeta^{[1]} \\
&+ \left(\frac{\partial f_\xi}{\partial x^{[1]}}\Big(\phi_\xi(\zeta)\Big)\langle \gamma_\xi^{[1]}, \gamma_\zeta^{[2]} \rangle + \frac{\partial f_\xi}{\partial x^{[2]}}\Big(\phi_\xi(\zeta)\Big)\langle \gamma_\xi^{[2]}, \gamma_\zeta^{[2]} \rangle\right)\gamma_\zeta^{[2]}.
\end{aligned}$$

This last expression coincides with $\text{grad}_\zeta f(\zeta)$, since

$$\nabla f_\zeta(\phi_\zeta(\zeta)) = \nabla f_\xi(\phi_\xi(\zeta)) J_{\xi\zeta}(\zeta) = \nabla_\xi f(\zeta)\, [\langle \gamma_\xi^{[i]}, \gamma_\zeta^{[j]} \rangle]_{i,j=1,2}$$

by the chain rule and (3).

The formulas (5) and (6) for the projection and inverse projection, respectively, follow immediately. □

2.3 Consistent Triangulations

Given a finite set $\mathcal{V}$ of points in Ω, let $\mathcal{T}$ be a set of triples $\tau = \{v, u, w\}$ of points $v, u, w \in \mathcal{V}$ such that

- any two triples have at most two common points,
- any pair of points in $\mathcal{V}$ belong to at most two different triples in $\mathcal{T}$, and
- for any $v \in \mathcal{V}$, the set of all triples containing v forms a *cell*, i.e. $\{\tau \in \mathcal{T} : v \in \tau\} = \{\tau_i : i = 1, \dots, n\}$ for some $n \geq 3$, where $\tau_i = \{v, v_i, v_{i+1}\}$, with $v_1, \dots, v_n$ all different, and $v_{n+1} = v_1$.

If these conditions are satisfied, we say that $\mathcal{T}$ is a *triangulation* of Ω with *vertices* $\mathcal{V}$. We say that two vertices v_1, v_2 are *connected* in $\mathcal{T}$ if there is a triple $\tau \in \mathcal{T}$ containing both v_1 and v_2. This definition of a triangulation of a manifold Ω is described by connectivity of vertices only, and does not involve 'edges' or 'triangles' on Ω. Indeed, $\mathcal{T}$ is essentially an *abstract simplicial complex* [21] with vertices in Ω.

Given $\xi \in \Omega$, assuming that all vertices of $\tau = \{v, u, w\} \in \mathcal{T}$ are in U_ξ, we denote by $\phi_\xi(\tau)$ the (open) planar triangle in B_ξ with vertices $\phi_\xi(u), \phi_\xi(v), \phi_\xi(w)$. Note that the triangle $\phi_\xi(\tau)$ may be degenerate. We set $\mathcal{T}_\xi = \{\tau \in \mathcal{T} : \tau \subset U_\xi\}$ and $\triangle_\xi = \{\phi_\xi(\tau) : \tau \in \mathcal{T}_\xi\}$.

Definition 1. A triangulation $\mathcal{T}$ of Ω is said to be *consistent* with the projection atlas $\Phi = \{(U_\xi, \phi_\xi)\}_{\xi \in \Omega}$ if for any $\xi \in \Omega$,

- every triangle $T \in \triangle_\xi$ is non-degenerate,
- $\triangle_\xi$ is a planar triangulation of $P_\xi := \cup_{T \in \triangle_\xi} \overline{T}$,
- $\phi_\xi(\xi)$ lies in the interior of P_ξ.

For any $\xi \in \Omega$, let $\mathcal{V}_\xi$ be the set consisting of vertices of all $\tau \in \mathcal{T}_\xi$ such that $\phi_\xi(\xi)$ lies in the closure of $\phi_\xi(\tau)$, i.e.

$$\mathcal{V}_\xi := \{v \in \mathcal{V} \cap U_\xi : \phi_\xi(\xi) \in \overline{\phi_\xi(\tau)} \text{ for some } \tau \in \mathcal{T}_\xi \text{ with a vertex at } v\}.$$

For a consistent triangulation $\mathcal{T}$, it is not difficult to check that $\mathcal{V}_\zeta \subseteq \mathcal{V}_\xi$ for all $\zeta \in \Omega$ sufficiently close to ξ, see [7]. Moreover, if ξ is a vertex in $\mathcal{V}$, then $\mathcal{V}_\xi$ consists of ξ and all vertices connected to it. For any point $\xi \in \Omega \setminus \mathcal{V}$, the set $\mathcal{V}_\xi$ contains either three or four points, depending on whether $\phi_\xi(\xi)$ belongs to the interior of a triangle in $\triangle_\xi$ or it lies on a common edge of two such triangles.

In addition to consistency, we will need the following assumption, specifically related to the Powell-Sabin spline:

$$\text{for every } \xi \in \Omega, \text{ if } \phi_\xi(\xi) \in \overline{T} \text{ for a } T \in \triangle_\xi, \text{ then } \triangle_\xi \text{ also includes three triangles sharing edges with } T. \tag{7}$$

We extend $\mathcal{V}_\xi$ to $\tilde{\mathcal{V}}_\xi$ by adding to $\mathcal{V}_\xi$ the vertices of the triangles described in (7).

3 Interpolation and Data Fitting

3.1 An Interpolation Method

Let $\mathcal{T}$ be a consistent triangulation of Ω. We assume that $\mathcal{T}$ is fine enough for (7) to hold. Let $D := \{a_v, c_v\}_{v\in\mathcal{V}}$, where a_v are real numbers and c_v are 3-vectors in Γ_v. We now show how to construct a C^1 function $s_\mathcal{T}$ defined on Ω that satisfies the interpolation conditions

$$s_\mathcal{T}(v) = a_v, \quad \text{grad}_v s_\mathcal{T}(v) = c_v, \qquad \text{all } v \in \mathcal{V}. \tag{8}$$

Algorithm 1. *Given $\xi \in \Omega$, compute $s_\mathcal{T}(\xi)$:*

1. *Let $T := \langle w_1, w_2, w_3\rangle$ be a triangle in $\triangle_\xi$ such that $\phi_\xi(\xi) \in \overline{T}$, and let $T_1 := \langle w_4, w_3, w_2\rangle$, $T_2 := \langle w_5, w_1, w_3\rangle$, and $T_3 := \langle w_6, w_2, w_1\rangle$ be the three triangles in $\triangle_\xi$ sharing edges with T, see Figure 1(left).*
2. *Let T_{PS} be the Powell-Sabin split of T into six triangles obtained by connecting the incenter w of T to the incenters of T_1, T_2, T_3, and to the vertices w_1, w_2, w_3, see Figure 1(right).*
3. *Let $g_i := c_{v_i} - \frac{\langle c_{v_i}, n_\xi\rangle}{\langle n_{v_i}, n_\xi\rangle} n_{v_i}$, where $v_i = \phi_\xi^{-1}(w_i)$, for $i = 1, 2, 3$.*
4. *Let $s_\mathcal{T}(\xi)$ be the value at $\phi_\xi(\xi)$ of the Powell-Sabin C^1 quadratic spline s_ξ defined on T_{PS} that interpolates the values $\{a_{v_i}\}_{i=1}^3$ and the gradients corresponding to $\{g_i\}_{i=1}^3$ at the vertices $\{w_i\}_{i=1}^3$, see* [26].

Since the Powell-Sabin interpolant in step 4 is uniquely defined by the values $\{(a_{v_i}, g_i)\}_{i=1}^3$ at the vertices $\{w_i\}_{i=1}^3$, it follows that $s_\mathcal{T}$ is uniquely defined by the data D. By construction, $s_\mathcal{T}$ satisfies (8). It would be tempting to consider $s_\mathcal{T}$ to be a spline on a partition of the manifold Ω obtained by drawing curves on Ω between connected vertices of $\mathcal{T}$. Indeed, $s_\mathcal{T}$ possesses a kind of piecewise structure, see [7]. However, this does not seem to be of a practical significance. What is important is that $s_\mathcal{T}$ is a smooth function.

Theorem 1 ([7]). *The interpolant $s_\mathcal{T}$ defined by Algorithm 1 is a C^1 function on the manifold Ω.*

Suppose that $s_\mathcal{T}(f)$ is the interpolant $s_\mathcal{T}$ corresponding to the data

$$a_v := f(v), \quad c_v := \text{grad}_v f(v), \qquad \text{all } v \in \mathcal{V},$$

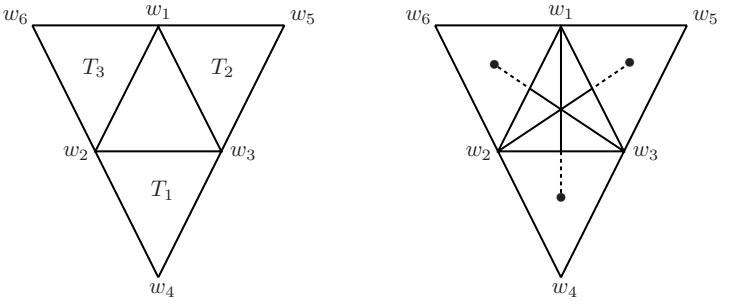


Fig. 1. The triangles in Algorithm 1

where f is a smooth function defined on Ω. Denote by h the *mesh size* of $\mathcal{T}$, by which we mean in the current setting the maximum distance in $\mathbb{R}^3$ between any pair of vertices $v \in \mathcal{V}$ connected in $\mathcal{T}$. By actually connecting these pairs of vertices by straight line segments, we obtain a 2-dimensional triangulation in $\mathbb{R}^3$. Let α be the *smallest angle* appearing in its triangles.

Theorem 2 ([7]). *If f is a sufficiently smooth real function on Ω, then*

$$\|f - s_{\mathcal{T}}(f)\|_{C(\Omega)} \le K\, h^3, \tag{9}$$

where K is a constant depending only on f and α.

Note that this error bound crucially depends on using the above formulas for the transformation of the gradients in step 3 of Algorithm 1 and in step 1(d) of Algorithm 2. These formulas are related to (6) and (5), respectively.

3.2 A Two-Stage Data Fitting Method

In practice we are frequently given only values of an unknown function f at a set X of scattered data points on the manifold Ω. In this case we can use a two-stage method to construct an approximation. First we select a consistent triangulation $\mathcal{T}$ of Ω satisfying (7). Let $\mathcal{V}$ be the set of vertices of $\mathcal{T}$. Note that we do not require that the vertices be located at the data points of X, and the number of vertices may be much smaller than the number of data points.

In the *first stage* of the algorithm we compute approximations to the values $\{f(v), \mathrm{grad}_v f(v)\}_{v\in\mathcal{V}}$ based on the data $\{f(\xi)\}_{\xi\in X}$. We perform these calculations in the sets $B_v := \phi_v(U_v) \subset \mathbb{R}^2$ using techniques available for local fitting of bivariate data. To carry this out, we suppose that

$$X \text{ is sufficiently dense so that } X \cap U_v \neq \emptyset \text{ for each } v \in \mathcal{V}. \tag{10}$$

Experience with the bivariate case [8] suggests that for each $v \in \mathcal{V}$, we compute both $a_v \approx f(v)$ and $c_v \approx \mathrm{grad}_v f(v)$ by averaging several estimates of the same quantities based on different sets of nearby data. It follows from the consistency of $\mathcal{T}$ that for each vertex $v \in \mathcal{V}$, all vertices of $\mathcal{T}$ connected to v belong to the set U_v.

Algorithm 2. *Given* $\{f(\xi)\}_{\xi\in X}$*, compute* $\{a_v, c_v\}_{v\in\mathcal{V}}$*:*

1. *For each* $v \in \mathcal{V}$*,*
   (a) *Let* $v_0 := v$*, and let* $v_1, v_2, \ldots, v_n \in \mathcal{V}$ *be the set of all vertices of* $\mathcal{T}$ *connected to* v*. Let* $\tilde{v}_i = \phi_v(v_i)$*,* $i = 1, \ldots, n$*.*
   (b) *Choose a set* $\tilde{X}_v \subset \phi_v(X \cap U_v)$ *of points in* B_v *near* $\phi_v(v)$*.*
   (c) *Compute a bivariate approximation* p_v *defined on* B_v *based on the data* $\{f_v(\tilde{\xi})\}_{\tilde{\xi}\in\tilde{X}_v}$*, where* $f_v := f \circ \phi_v^{-1}$*.*
   (d) *Store the numbers* $a_{v,v_i} := p_v(\tilde{v}_i)$ *and vectors* $c_{v,v_i} := \operatorname{grad} p_v(\tilde{v}_i) - \langle \operatorname{grad} p_v(\tilde{v}_i), n_{v_i}\rangle n_{v_i}$ *for* $i = 0, \ldots, n$*.*
2. *For each* $v \in \mathcal{V}$*, set*

$$a_v := \frac{1}{n+1}\sum_{i=0}^{n} a_{v_i,v}, \quad c_v := \frac{1}{n+1}\sum_{i=0}^{n} c_{v_i,v}.$$

In the *second stage* of the algorithm we construct our approximant $s_{\mathcal{T}}$ as the interpolant (8) to the data $\{a_v, c_v\}_{v\in\mathcal{V}}$ obtained from Algorithm 2.

We have not specified how $\mathcal{T}$ is selected and how the steps 1(b) and 1(c) of Algorithm 2 are to be performed. However, the overall performance of the two-stage method will depend significantly on the particular techniques used in these steps. We discuss two numerical examples in Section 4, using recently developed adaptive techniques based on local least squares fitting by bivariate polynomials and radial basis functions [5,6,8].

We now give an error bound for this method in terms of the mesh size h and the approximation power of the local approximations p_v.

Let $\kappa(\xi)$ be the minimum of $\langle n_\xi, n_v\rangle$ over all $v \in \mathcal{V} \cap U_\xi$ such that $\xi = \phi_\xi(\xi)$ belongs to the closure of a triangle of $\triangle_\xi$ attached to $\phi_\xi(v)$, and let

$$\kappa = \min_{\xi\in\Omega} \kappa(\xi).$$

By the definition of U_ξ, it follows that $\kappa > 0$. Moreover, $\kappa \to 1$ as the mesh size h of $\mathcal{T}$ goes to 0.

For each $v \in \mathcal{V}$, let N_v be the union of all triangles of $\triangle_v$ attached to v, and let p_v be the bivariate approximation to $f_v = f \circ \phi_v^{-1}$, as in Algorithm 2.

Theorem 3 ([7]). *Let* $s_{\mathcal{T}}$ *be the approximant to a sufficiently smooth function* f *on* Ω *constructed by the above two-stage scattered data fitting method. Then*

$$\|f - s_{\mathcal{T}}\|_{C(\Omega)} \le K\Big[C_f h^3 + \max_{v\in\mathcal{V}}\big\{\|f_v - p_v\|_{C(N_v)} + h\|\operatorname{grad} f_v - \operatorname{grad} p_v\|_{C(N_v)}\big\}\Big],$$

where the constant K *depends only on* κ *and the smallest angle* α*, and* C_f *depends only on* f*.*

Clearly, this theorem shows that if the local approximations p_v are powerful enough to guarantee an $\mathcal{O}(h^3)$ error, then the overall error of the two-stage method is also $\mathcal{O}(h^3)$. This asymptotic behavior of the error is confirmed by the numerical examples in the next section.

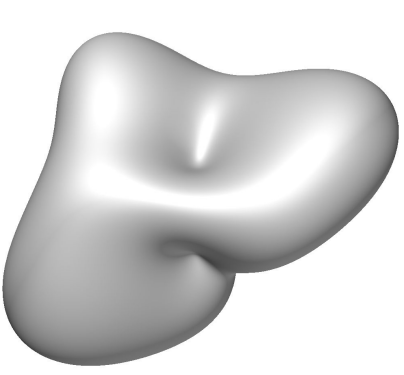


Fig. 2. Test function f_1 on the sphere

4 Numerical Examples

4.1 Scattered Data Fitting on the Unit Sphere

For our first example, we choose the manifold Ω to be the unit sphere in $\mathbb{R}^3$ defined by $x^2 + y^2 + z^2 = 1$. As a test function, we take the function

$$f_1(x, y, z) := 1 + x^8 + e^{2y^3} + e^{2z^2} + 10xyz$$

used in the examples in [1]. As in [1], we visualize this function as a kind of offset surface to the manifold, i.e., we plot

$$\{\xi + f_1(\xi)n_\xi \,:\, \xi \in \Omega\},$$

where n_ξ denotes the unit outer normal to the manifold Ω at ξ, see Figure 2.

In order to study the behavior of the error as a function of mesh size, we use the nested sequence of triangulations introduced in [1]. Let $\mathcal{T}_0$ be the regular octahedron with vertices at $\pm e_i$, $i = 1, 2, 3$, where e_i are the Cartesian coordinate vectors. This triangulation has 6 vertices and 8 triangles. We now define $\mathcal{T}_n$ by repeated refinement, where $\mathcal{T}_n$ is obtained from $\mathcal{T}_{n-1}$ by adding vertices at the midpoints of the great circle arcs connecting neighboring vertices of $\mathcal{T}_{n-1}$, and then splitting each triangle in $\mathcal{T}_{n-1}$ into four subtriangles using these new vertices. The number of vertices of $\mathcal{T}_n$ is $V_n = 2^{2n+2} + 2$, and the number of triangles is 2^{2n+3}. The triangulations $\mathcal{T}_1$ and $\mathcal{T}_2$ are shown in Figure 3. Note that $\mathcal{T}_0$ is not sufficiently fine for our Powell-Sabin interpolant to be defined, see Section 3.1.

To get test data, we use a simple spherical random number generator, see Remark 6. In particular, for each $n = 1, 2, \ldots, 6$, we generate $3V_n = 3(2^{2n+2}+2)$ random points. We choose this number since as shown in Section 3.1, the Powell-Sabin interpolant is uniquely defined by $3V_n$ degrees of freedom. We then evaluate the test function f_1 at these points, and create approximants s_n by the two-stage method described in Section 3.2.

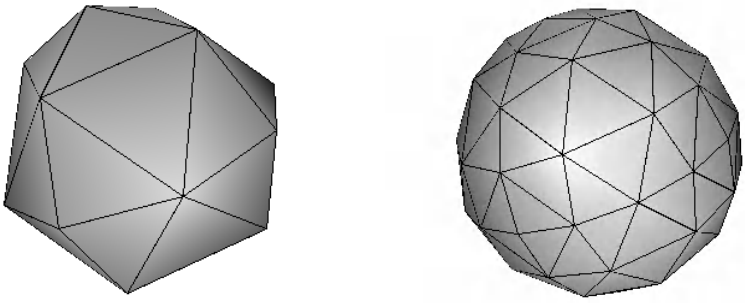


Fig. 3. Triangulations $\mathcal{T}_1$ and $\mathcal{T}_2$

In the first stage we use least squares multiquadric fitting as described in [6] with the following parameter values: minimum number of points $M_{min} = 25$, maximum number of points $M_{max} = 100$, separation $S = 35$, and scaling $\delta = 0.2$ if $n = 1$, $\delta = 1.0$ if $n = 2$, $\delta = 2.0$ if $n = 3$, $\delta = 3.0$ if $n \in \{4, 5\}$, $\delta = 4.0$ if $n = 6$. See [6] for the exact meaning of M_{min}, M_{max}, S and δ. The choice of increasing values for the δ's as the number of data points increases is motivated by numerical results in [6]. In our implementation we make use of the corresponding subroutines from the software library TSFIT [9].

The results of our experiments are presented in Table 1. In the column labelled *Two Stage*, we list the relative maximum errors $\|f_1 - s_n\|_\infty / \|f_1\|_\infty$ for $n = 1, \ldots, 6$. For comparison purposes, in the column labelled *Exact*, we list the relative errors when the exact function values and projected gradients are used instead of the approximate values obtained from first stage fitting. In the column labelled *Spherical PS* we also list the relative errors corresponding to using the spherical Powell Sabin interpolant of [1] based on the exact function values and gradients at the vertices of $\mathcal{T}_n$.

The table shows that the errors for the three methods are comparable, and indeed for $n \geq 2$ are almost identical. To test the rate of convergence, in the column labelled *Ratio* we list the ratios of the errors of our two-stage method, i.e., $\|f_1 - s_{n-1}\| / \|f_1 - s_n\|$ for $n = 2, \ldots, 6$. Since the mesh size decreases by approximately $1/2$ at each refinement step, and since the error should be of size $\mathcal{O}(h^3)$, these ratios tend to 8 as they should.

To illustrate the performance of the two-stage method with even fewer data points, we recomputed the spline fit corresponding to the triangulation $\mathcal{T}_2$, but with only 100 random points on the sphere. The parameters for the first step were taken to be $M_{min} = 25$, $M_{max} = 100$, $S = 25$ and $\delta = 0.8$. In this case the maximum relative error was 2.18×10^{-2}. Figure 4 shows the test function f_1 along with the data sites and the approximant computed using the two-stage method.

Table 1. Tests with random data on the sphere

	# data	Two Stage	Ratio	Exact	Spherical PS
$\mathcal{T}_1$	54	2.74×10^{-1}		8.08×10^{-2}	7.83×10^{-2}
$\mathcal{T}_2$	198	2.15×10^{-2}	12.7	2.17×10^{-2}	2.10×10^{-2}
$\mathcal{T}_3$	774	2.10×10^{-3}	10.2	2.10×10^{-3}	2.05×10^{-3}
$\mathcal{T}_4$	3078	2.41×10^{-4}	8.7	2.41×10^{-4}	2.28×10^{-4}
$\mathcal{T}_5$	12294	2.92×10^{-5}	8.3	2.92×10^{-5}	2.88×10^{-5}
$\mathcal{T}_6$	49158	3.65×10^{-6}	8.0	3.65×10^{-6}	3.60×10^{-6}

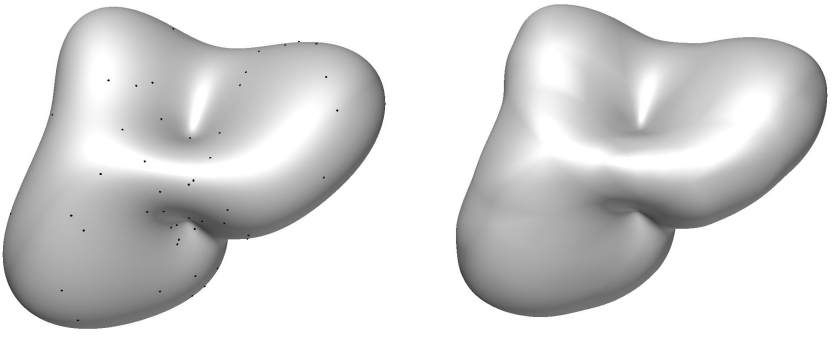


Fig. 4. Test function with 100 data sites (left) and its approximation (right)

4.2 Ring Type Surfaces

As a second example, we consider the ring-type surfaces used in the examples presented in [25]. Given a real number $0 \le a < 1$ and an integer $m \ge 0$, we define a smooth 2-manifold in $\mathbb{R}^3$ parametrically via

$$x = [2 + (1 + a\cos mu)\cos v]\cos u, \tag{11}$$

$$y = [2 + (1 + a\cos mu)\cos v]\sin u, \tag{12}$$

$$z = [1 + a\cos mu]\sin v, \tag{13}$$

where (u, v) runs over the parameter domain $[0, 2\pi) \times [0, 2\pi)$. When $m = 0$, this corresponds to a torus with outer radius $3 + a$ and inner radius $1 - a$. In general it is a surface of genus 1.

For our experiments, we choose $a = 0.3$ and $m = 5$, giving rise to the manifold depicted in Figure 5 (left). As a test function, we now take

$$f_2(x, y, z) := (1 + x^8 + y^3 + z^2)/4000.$$

The corresponding surface is shown in Figure 5 (right).

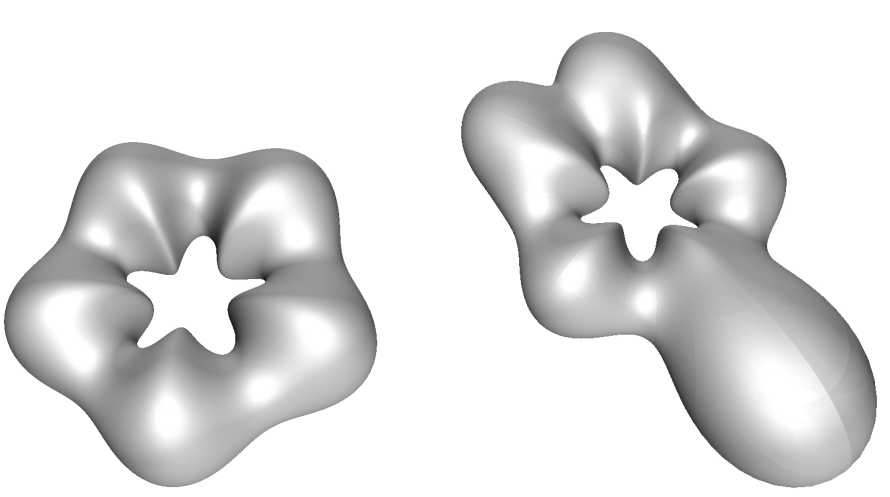


Fig. 5. Ring-type surface with $a = 0.3$ and $m = 5$ (left), and the test function f_2 visualized as an offset surface (right)

For comparison purposes, we generate a sequence of triangulations $\mathcal{T}_n$, $n = 1, \ldots, 5$, by starting with nested three directional meshes in the parameter domain $[0, 2\pi) \times [0, 2\pi)$ with vertices (u_i, v_j) given by

$$u_i = \frac{2\pi i}{24 \cdot 2^{n-1}}, \qquad v_j = \frac{2\pi j}{15 \cdot 2^{n-1}},$$

and mapping these vertices onto the ring-type surface. The triangulations $\mathcal{T}_1$ and $\mathcal{T}_2$ are shown in Figure 6. The number of vertices of $\mathcal{T}_n$ is $V_n = 90 \cdot 4^n$, and the number of triangles is $180 \cdot 4^n$. To generate data for our experiments, we evaluate the test function f_2 at $3V_n$ random points on the surface. As in

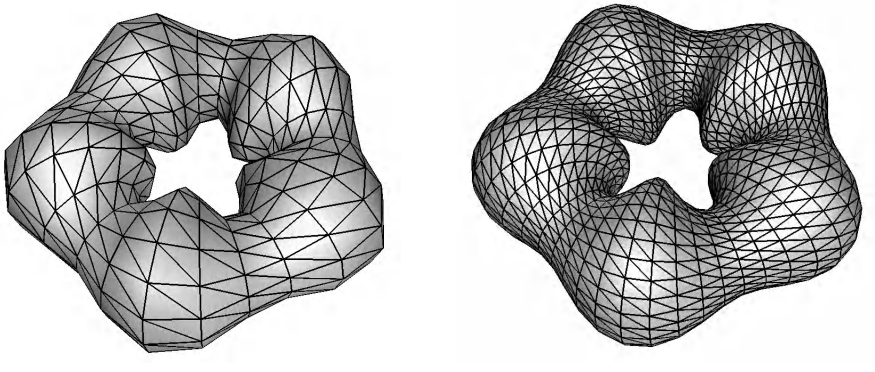


Fig. 6. Triangulations $\mathcal{T}_1$ and $\mathcal{T}_2$

Table 2. Tests with random data on the ring type surface

	# data	Two Stage	Ratio	Exact
$\mathcal{T}_1$	1080	5.57×10^{-2}		5.50×10^{-2}
$\mathcal{T}_2$	4320	2.84×10^{-2}	1.96	2.84×10^{-2}
$\mathcal{T}_3$	17280	4.01×10^{-3}	7.08	4.01×10^{-3}
$\mathcal{T}_4$	69120	9.18×10^{-5}	43.7	9.17×10^{-5}
$\mathcal{T}_5$	276480	1.14×10^{-5}	8.05	1.14×10^{-5}

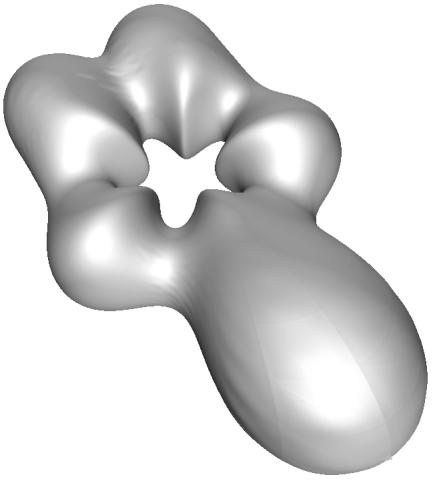


Fig. 7. Powell-Sabin approximation computed using 1000 data points

Section 4.1, we use local multiquadric fitting [6] with the following parameter values: $M_{min} = 25$, $M_{max} = 100$, $S = 15$, and $\delta = 2.0$ if $n = 1$, $\delta = 3.0$ if $n \geq 2$. Table 2 shows the relative maximum error for the two-stage method and for the exact interpolant. The somewhat irregular convergence rate seen in the 'ratio' column is probably due to the fact that we are using easy to generate triangulations that are not well adapted to this particular manifold.

Figure 7 shows the $\mathcal{T}_2$-approximation computed using 1000 random data, with parameters $M_{min} = 25$, $M_{max} = 100$, $S = 15$, and $\delta = 2.0$. The relative error of this approximation is 2.89×10^{-2}.

5 Remarks

Remark 1. The problem of fitting functions defined on surfaces arises in many applications, see [1,2,3,4,10,11,12,13,14,19,20,23,25,27,29], and references therein. Used parametrically, such functions can be applied to the problem of modelling surfaces of arbitrary topological type from point clouds, see [15,16,30].

Remark 2. Many of the papers mentioned in the above remark deal with the sphere in $\mathbb{R}^3$. For a survey of interpolation and scattered data fitting methods on the sphere, see [12]. For some specific methods, see [14,19,20,22,23,27,29].

Remark 3. The method of this paper is closely related to work of Demjanovich [10,11]. He also computes an interpolant s at a point ξ on the manifold by projecting into the tangent plane and using a bivariate finite element interpolation method. A key difference is that for each evaluation point ξ, his method involves interpolation of the original function at the points in the tangent plane Γ_ξ determined by the finite element scheme, whereas in our methods we only interpolate projected gradients corresponding to the vertices of the underlying triangulation $\mathcal{T}$, compare steps 3 and 4 of Algorithm 1. Therefore, our interpolation operator only requires function values and gradients at the vertices of $\mathcal{T}$, which makes it possible to design a two-stage scattered data fitting method. In [10,11] only interpolation with Courant hat functions has similar properties for general manifolds, but it does not produce a C^1 interpolant.

Remark 4. The method of this paper is also closely related to work of Pottmann [25], which also makes use of projected gradients. (It is not difficult to see that our equation (6) describes the π-*transform* of [25].) However, instead of using local approximation methods to estimate gradients, he constructs a kind of minimum norm network.

Remark 5. Here we have made use of the standard bivariate C^1 quadratic Powell-Sabin macro-element to solve the interpolation problem in the tangent plane. Its key feature is that it is constructed from only nine pieces of data, the values and gradients at the three vertices of the macro-triangle. Using the same data, we can also construct an interpolant based on the classical C^1 *reduced Clough-Tocher macro-element*. It is based on a split of the macro-triangle into three subtriangles (typically using the barycenter), and is a cubic polynomial on each piece. Along each edge its cross derivative is restricted to be a linear polynomial. Yet another possibility is a modified quadratic *Powell-Sabin macro-element on a 12-split* [26], where the cross derivatives are assumed linear rather than piecewise linear on the edges of the macro-triangles. Note that with either a Clough-Tocher or 12-split Powell-Sabin macro-element the assumption (7) will not be needed.

Remark 6. To simulate scattered data for our numerical experiments in Section 4, we generated pseudo-random points uniformly distributed on the test surfaces. For the *surface of the sphere* we use the following method described e.g. in [24]: To obtain a point (x, y, z) on the unit sphere, generate two pseudo-random real numbers z and t uniformly distributed in $[-1, 1]$ and $[0, 2\pi)$, respectively, and then compute $x = \rho\cos t$ and $y = \rho\sin t$, where $\rho = \sqrt{1 - z^2}$. For the *ring-type surface* defined parametrically by (11)–(13), the points $r(u, v) = (x(u, v), y(u, v), z(u, v))$ will be uniformly distributed on the surface if $(u, v) \in [0, 2\pi) \times [0, 2\pi)$ are chosen according to the probability distribution $p(u, v) = \alpha\|r_u \times r_v\|_2$, with $\alpha \in \mathbb{R}$ such that $\int_{[0,2\pi)\times[0,2\pi)} p(u, v)\, du\, dv = 1$. It is not difficult to see that $\|r_u \times r_v\|_2^2 = \psi^2[(\psi')^2 + (2 + \psi\cos v)^2]$, where $\psi(u) = 1 + a\cos mu$.

Using this explicit formula, we employ the well-known von Neumann rejection method to generate the points.

Acknowledgments

The first named author was partially supported by the Edinburgh Mathematical Society Research Support Fund.

References

1. Alfeld, P., Neamtu, M., Schumaker, L.L.: Fitting Scattered Data on Sphere-like Surfaces using Spherical Splines. J. Comp. Appl. Math. 73, 5–43 (1996)
2. Barnhill, R.E., Foley, T.A.: Methods for Constructing Surfaces on Surfaces. In: Geometric Modeling: Methods and Their Applications, pp. 1–15. Springer, Heidelberg (1991)
3. Barnhill, R.E., Piper, B.R., Stead, S.E.: A Multidimensional Surface Problem: Pressure on a Wing. Comput. Aided Geom. Design 2, 185–187 (1985)
4. Barnhill, R.E., Ou, H.S.: Surfaces Defined on Surfaces. Comput. Aided Geom. Design 7, 323–336 (1990)
5. Davydov, O., Morandi, R., Sestini, A.: Local Hybrid Approximation for Scattered Data Fitting with Bivariate Splines. Comput. Aided Geom. Design 23, 703–721 (2006)
6. Davydov, O., Sestini, A., Morandi, R.: Local RBF Approximation for Scattered Data Fitting with Bivariate Splines. In: de Bruin, M.G., Mache, D.H., Szabados, J. (eds.) Trends and Applications in Constructive Approximation, ISNM, vol. 151, pp. 91–102, Birkhäuser (2005)
7. Davydov, O., Schumaker, L.L.: Interpolation and Scattered Data Fitting on Manifolds using Projected Powell-Sabin Splines. Manuscript (2007)
8. Davydov, O., Zeilfelder, F.: Scattered Data Fitting by Direct Extension of Local Polynomials to Bivariate Splines. Advances in Comp. Math. 21, 223–271 (2004)
9. Davydov, O., Zeilfelder, F.: TSFIT: A Software Package for Two-Stage Scattered Data Fitting (2005), Available under GPL from `http://www.maths.strath.ac.uk/~aas04108/tsfit/`
10. Demjanovich, Y.K.: Construction of Spaces of Local Functions on Manifolds. Metody Vychisl. 14, 100–109 (1985)
11. Demjanovich, Y.K.: Local Approximations of Functions Given on Manifolds. Amer. Math. Soc. Transl. 159(2), 53–76 (1994)
12. Fasshauer, G., Schumaker, L.L.: Scattered Data Fitting on the Sphere. In: Dæhlen, M., Lyche, T., Schumaker, L.L. (eds.) Mathematical Methods for Curves and Surfaces II, pp. 117–166. Vanderbilt University Press, Nashville (1998)
13. Foley, T.A., Lane, D.A., Nielson, G.M., Franke, R., Hagen, H.: Interpolation of Scattered Data on Closed Surfaces. Comput. Aided Geom. Design 7, 303–312 (1990)
14. Freeden, W.: Spherical Spline Interpolation—Basic Theory and Computational Aspects. J. Comp. Appl. Math. 11, 367–375 (1985)
15. Grimm, C.M., Hughes, J.F.: Modeling Surfaces of Arbitrary Topology Using Manifolds. In: Proc. SIGGRAPH 95, pp. 359–368 (1995)

16. Gu, X., He, Y., Qin, H.: Manifold Splines. In: Proc ACM Symp. Solid and Physical Modeling, pp. 27–38 (2005)
17. Hirsch, M.W.: Differential Topology. Springer, Heidelberg (1976)
18. Lai, M.-J., Schumaker, L.L.: Spline Functions on Triangulations. Cambridge University Press, Cambridge (2007)
19. Lawson, C.L.: C^1 Surface Interpolation for Scattered Data on a Sphere. Rocky Mountain J. Math. 14, 177–202 (1984)
20. Lyche, T., Schumaker, L.L.: A Multiresolution Tensor Spline Method for Fitting Functions on the Sphere. SIAM J. Sci. Computing 22, 724–746 (2000)
21. Maunder, C.R.F.: Algebraic Topology. Van Nostrand Reinhold, London (1970)
22. Neamtu, M., Schumaker, L.L.: On the Approximation Order of Splines on Spherical Triangulations. Adv. in Comp. Math. 21, 3–20 (2004)
23. Nielson, G.M., Ramaraj, R.: Interpolation over a Sphere Based upon a Minimum Norm Network. Comput. Aided Geom. Design 4, 41–58 (1987)
24. O'Rourke, J.: Computational Geometry in C, 2nd edn. Cambridge University Press, Cambridge (1998)
25. Pottmann, H.: Interpolation on Surfaces using Minimum Norm Networks. CAGD 9, 51–67 (1992)
26. Powell, M.J.D., Sabin, M.A.: Piecewise Quadratic Approximations on Triangles. ACM Trans. Math. Software 3, 316–325 (1977)
27. Schumaker, L.L., Traas, C.: Fitting Scattered Data on Spherelike Surfaces using Tensor Products of Trigonometric and Polynomial Splines. Numer. Math. 60, 133–144 (1991)
28. Thorpe, J.A.: Elementary Topics in Differential Geometry. Springer, Heidelberg (1979)
29. Wahba, G.: Spline Interpolation and Smoothing on the Sphere. SIAM J. Sci. Stat. Computing 2, 5–16 (1981)
30. Ying, L., Zorin, D.: A Simple Manifold-based Construction of Surfaces of Arbitrary Smoothness. ACM Transactions on Graphics 23, 271–275 (2004)

Implicit Boundary Control of Vector Field Based Shape Deformations

Wolfram von Funck[1], Holger Theisel[2], and Hans-Peter Seidel[3]

[1] MPI Informatik, D-66123 Saarbrücken, Germany
wfunck@mpi-inf.mpg.de
http://www.mpi-inf.mpg.de/~wfunck
[2] Computer Graphics Group, Bielefeld University, D-33501 Bielefeld, Germany
[3] MPI Informatik, D-66123 Saarbrücken, Germany

Abstract. We present a shape deformation approach which preserves volume, prevents self-intersections and allows for exact control of the deformation impact. The volume preservation and prevention of self-intersections are achieved by utilizing the method of Vector Field Based Shape Deformations. This method produces physically plausible deformations efficiently by integrating formally constructed divergence-free vector fields, where the region of influence is described by implicitly defined shapes. We introduce an implicit representation of deformation boundaries, which allows for an exact control of the deformation: By placing the boundaries directly on the shape surface, the user can specify precisely where the shape should be deformed and where not. The simple polygonal representation of the boundaries allows for a GPU implementation, which is able to deform high-resolution meshes in real-time.

1 Introduction

Deforming shapes under a number of constraints is a standard problem in Computer Graphics. For instance, animation of characters can be achieved by deforming the shape of the character according to its underlying skeleton, elastic bodies are deformed by performing simulations based on physical laws, or in industrial design, fair surface deformations are obtained by minimizing curvature energy.

When deforming solid objects, the constraint of volume preservation is an important and often-addressed issue: Under the assumption that the object consists of an incompressible material, its volume remains constant under deformation. While physical simulations or constrained optimization techniques can be used to achieve this goal, they are usuallycomputationally expensive and require special data structures like grids or embedding meshes. In contrast to this, the method of *Vector Field Based Shape Deformations* (VFSD) [1] constructs and integrates divergence-free vector fields on-the-fly without any simulation, optimization or special data structures and produces realistic looking volume-preserving deformations without self-intersections.

Boundary constraints, often used in the context of shape editing, are another useful aspect of shape deformations. The user often wants to specify precisely

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 154–165, 2007.
© Springer-Verlag Berlin Heidelberg 2007

which parts of the shape should be deformed and which parts should not be deformed at all. This is usually done by placing boundary constraints on the surface, i.e. the user draws two curves on the surface. The region enclosed by the first curve undergoes a full deformation, e.g. a translation or rotation. The region enclosed by the second curve is not deformed at all. In the region between both curves, the deformation is smoothly blended between full and zero deformation.

In this paper, we present a method which brings both approaches together. The user can define the deformation impact by drawing two boundaries onto the surface of the shape and can deform the shape in a volume-preserving and foldover-free manner with respect to these boundaries. While in most existing approaches boundaries are constraints of an optimization, we introduce implicit boundaries, which are defined by closed polygons and give a direct mathematical solution of a smooth blending function which defines the amount of deformation for every point in space.

The paper is organized as follows: Section 2 reviews relevant shape deformation approaches. Section 3 describes how implicit boundaries are defined and how they can be used together with Vector Field Based Shape Deformations to deform triangle meshes. Section 4 demonstrates several application scenarios, while Section 5 goes into the details of the implementation and analyzes the performance. Finally, Section 6 discusses the presented method and possible future research.

2 Related Work

We would like to give the reader an overview of related work. We review both surface-based approaches and space deformations, since our approach is a space deformation which is constructed by defining boundaries and the surface of the shape.

Surface based methods define the deformation on the surface of the shape. A common approach, based on triangle meshes, is to specify a number of original and target vertices and compute the remaining vertex positions by a variational approach [2,3]. Boundary constraint-modeling has established itself as a standard method for deforming surfaces represented by triangle meshes. The main idea is that the user specifies boundary constraints as displacements of a number of vertices. Usually this means that two bands of vertices, the boundaries, are marked on the surface, where on one band the displacements are zero and on the other band the displacements describe a simple deformation like translation, rotation or scaling. Using variational calculus, a deformation field which is optimal with respect to curvature energy is found for the free vertices [4,5,6,7,8,9,10]. This involves solving a sparse linear Laplacian system during each modification by the user.

Space deformations are defined for all points in space, i.e. a shape is deformed by deforming the space where it is placed in. The first space deformation methods appeared in the form of free-form deformations (FFD) [11]. The idea of FFD is to define deformations by modifying coarse control structures like

lattices [11,12,13], curves [14,15], or points [16,17]. Using radial basis functions, it is possible to extend the boundary constraint modeling method from the surface setting to the space setting [18]. In order to give the user the impression of real incompressible material, many space deformation approaches have been developed which preserve the volume of the shape under deformation [16,19,20,21,22,1]. Since self-intersections are irreversible using space deformations, a number of approaches have been developed to address this problem [23,24,25].

Another representative of space deformations, which addresses the issues of volume-preservation and prevention of self-intersections, is the method of Vector Field Based Shape Deformations [1]. This method relies on a formal construction of time-dependent divergence-free vector fields on which path line integrations are carried out to deform the vertices of a triangle mesh. Due to the C^1 continuity of the vector fields, the resulting deformation is smooth. Due to the path line integration, self-intersections are prevented. Since the vector fields are divergence-free, the volume of the shape remains constant under deformation. Thanks to the direct mathematical formulation of the vector fields, the deformation if independ of the shape representation, requires neither special control structures nor precomputations. In the following, we review the method. To simplify matters, we formulate the construction in a time-independent context – the extension to the time-dependent case is straightforward.

Given two C^2 continuous scalar fields $p, q : \mathbb{R}^3 \rightarrow \mathbb{R}$, a C^1 continuous divergence-free vector field $\mathbf{v}$ can be constructed from the gradients of p and q as

$$\mathbf{v}(x, y, z) = \nabla p(x, y, z) \times \nabla q(x, y, z). \tag{1}$$

Simple deformations can be constructed with this method using linear or quadratic scalar fields. In particular, a translation can be achieved by using two linear fields

$$e(\mathbf{x}) = \mathbf{u}(\mathbf{x} - \mathbf{c})^T, f(\mathbf{x}) = \mathbf{w}(\mathbf{x} - \mathbf{c})^T, \tag{2}$$

where $\mathbf{u}$ and $\mathbf{w}$ are orthogonal normalized vectors and $\mathbf{c}$ is an arbitrary center point. Since $\mathbf{u}$ and $\mathbf{w}$ are the gradients of e and f, respectively, $\mathbf{u} \times \mathbf{w}$ defines the translation direction. A rotational vector field can be constructed from a linear and a quadratic field:

$$e(\mathbf{x}) = \mathbf{a}(\mathbf{x} - \mathbf{c})^T, f(\mathbf{x}) = (\mathbf{a} \times (\mathbf{x} - \mathbf{c})^T)^2 \tag{3}$$

The rotation axis is defined by the normalized vector $\mathbf{a}$, while $\mathbf{c}$ describes the rotation center.

By performing a stream line integration (or path line integration for the time-dependent case) of each mesh vertex in the resulting vector fields, it is possible to rotate and translate a mesh arbitrarily. Obviously, such transformations can be achieved more easily and efficiently by other means. However, we can create more complex and local deformations by blending the scalar functions e, f using a third function b, the *blending function*. In [1], the blending is done in a piecewise manner, where b is the function of a distance field. Here, we define b more

generally as a C^2 continuous field $b : \mathbb{R}^3 \rightarrow [0, 1]$. Given b, we can define the blended fields

$$p(\mathbf{x}) = b(\mathbf{x}) \cdot e(\mathbf{x}) \tag{4}$$
$$q(\mathbf{x}) = b(\mathbf{x}) \cdot f(\mathbf{x}). \tag{5}$$

Using (2) for translations or (3) for rotations, we can insert (4) and (5) into (1) to obtain a divergence-free vector field, which describes

- a full translation/rotation at points where $b(\mathbf{x}) = 1$,
- a zero-deformation where $b(\mathbf{x}) = 0$,
- a smoothly blended deformation for points where $0 < b(\mathbf{x}) < 1$.

By defining an appropriate blending function b, it is possible to specify which points in space should be deformed by what amount. Figure 1 demonstrates this in the 2D setting. Since the blending is done before the cross product of the gradients is computed (Equation 1), the resulting vector field is still divergence-free. Since p and q are C^2 continuous, the resulting vector field $\mathbf{v}$ is C^1 continuous [1].

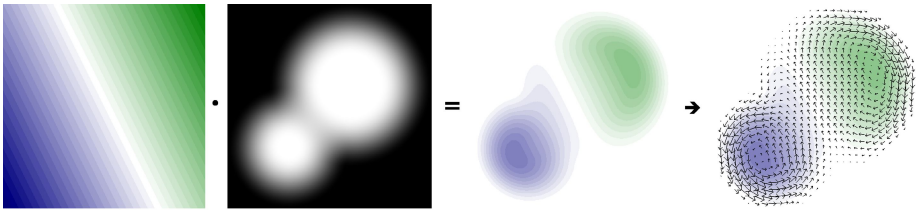


Fig. 1. Blending the deformation. A linear field (left), describing a translation along its isolines, is multiplied with a blending function (center left). The result is a blended field (center right) from which a divergence-free deformation field (right) can be computed.

3 Deformation Blending

In the following sections, we will show a method to construct a blending function which can be used for boundary constraint modeling based on VFSD.

3.1 Implicit Boundaries

From a technical point of view, we don't use the term *boundary constraints* in the sense of an optimization problem, but use boundary constraints as user defined positions in space where the surface should be deformed in a prescribed manner. From the user's point of view, our approach resembles other boundary constraint modeling approaches: the user draws two boundaries on the surface, where the outer boundary defines the support of the deformation, while the inner boundary defines the control handle of the deformation. Figure 2 illustrates this.

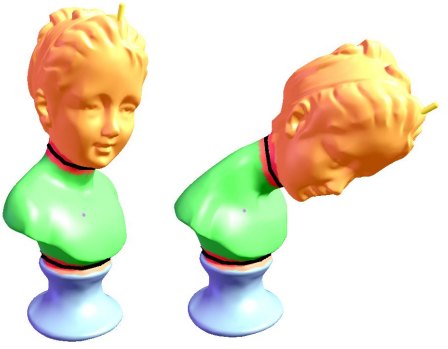


Fig. 2. By drawing boundaries onto the surface, the user can define a support region (here the body) and a handle region (head)

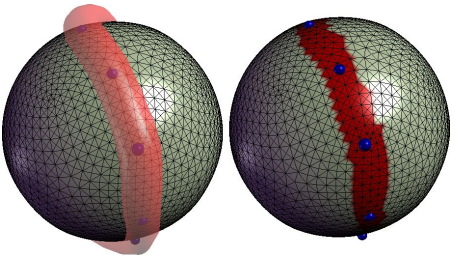


Fig. 3. Left: an implicit boundary defined by a few points. Right: the corresponding vertices marked.

Since VFSD is not a surface-based technique but describes space deformations, the boundaries have to be defined in space and not only on the surface. Therefore, we formulate them implicitly. We do so by constructing a smooth implicit function for both the inner and the outer boundary. More precisely, we define for each boundary a closed piecewise linear curve, i.e. a ring of connected line segments. Then we use an approximate smooth distance field to each curve as implicit function. Given n points $\mathbf{b}_j$, $1 <= j <= n$ with $\mathbf{b}_{n+1} := \mathbf{b}_1$, defining such a polygonal curve, we can compute an approximate distance field using the technique from [26] as follows: given the Euclidian distance fields $l_j(\mathbf{x})$ of each line segment defined by the endpoints $\mathbf{b}_j$, $\mathbf{b}_{j+1}$, we get

$$d(\mathbf{x}) = \frac{1}{\sqrt[k]{\sum_{i=1}^{n} \frac{1}{(l_i(\mathbf{x}))^k}}} \tag{6}$$

This corresponds to the *R-equivalence* $l_1(\mathbf{x}) \sim ... \sim l_n(\mathbf{x})$ described in [26], which joins the distance fields $l_j(\mathbf{x})$ to a smooth approximate one. k is a positive integer, which basically controls the "exactness" of the distance field: the greater k, the more the approximation approaches the real distance field of the polygonal curve, which, in general, contains discontinuities. The smaller we choose k, the smoother the approximation becomes. In our implementation, we use $k = 2$ in order to obtain smooth deformations even for coarse polygons.

Given such a scalar field for both the inner and the outer boundary, i.e. $d_i(\mathbf{x})$ and $d_o(\mathbf{x})$, we want to constrain the deformation as follows:

- if $d_i(\mathbf{x}) = t_i$, we want full deformation,
- if $d_o(\mathbf{x}) = t_o$, we want no deformation,
- else, we want a smooth blending between full and zero deformation.

t_i and t_o are user-adjustable thresholds which define the thickness of the innner and outer boundary, respectively. Visually, an implicit boundary can be seen as a closed tube with adjustable thickness running over the surface, as depicted in Figure 3. Besides the necessary implicit formulation, this has the advantage that the number of polygon vertices is independent of the mesh resolution, and the user can define smooth boundaries with only a few points, which is especially useful for parallel computation on the GPU. In order to avoid discontinuities in the deformation, the user has to avoid intersections between inner and outer boundary. For instance, when deforming the Dragon's mouth as in Figure 4, the boundary thickness has to be chosen such that the boundaries don't touch each other between upper and lower jaw. The points $\mathbf{b}_j$ and t_i, t_o have to be chosen such that the boundary area on the surface is connected, i.e., it divides the shape into two parts. For given t_i, t_o, this can always be achieved by increasing the density of the control points $\mathbf{b}_j$ and by placing them close to sharp features.

3.2 Smooth Blending

Having the implicit boundaries defined, we need to construct a function that can be used to blend smoothly from full deformation to zero deformation between inner and outer boundary. This can be accomplished in a straightforward way by interpolation with inverse distance weighting [27]. We define the blending function as

$$b(\mathbf{x}) = \frac{1/(d_i(\mathbf{x}) - t_i))^3 \cdot 1 + 1/(d_o(\mathbf{x}) - t_o))^3 \cdot 0}{1/(d_i(\mathbf{x}) - t_i))^3 + 1/(d_o(\mathbf{x}) - t_o))^3} \tag{7}$$

In the limit, the following holds: $b(\mathbf{x}) = 1$ for $d_i(\mathbf{x}) = t_i$ and $b(\mathbf{x}) = 0$ for $d_o(\mathbf{x}) = t_o$. Because of the cubic weights, $b(\mathbf{x})$ has two vanishing derivatives at points with $d_i(\mathbf{x}) = t_i$ or $d_o(\mathbf{x}) = t_o$. As we will see later, this is an important property which is needed to perform the deformation of the mesh in a piecewise manner. A further degree of freedom can be achieved by multiplying the weights with user-defined factors. This is especially useful to control bend deformations.

So far, we have a C^2 continuous blending function b which can be used together with Equations (4), (5) to construct blended scalar fields with (2) for translations or with (3) for rotations. Using (1), we can construct a divergence-free vector field, which deforms the mesh (nearly) according to the boundary constraints. Vertices on the inner boundary, i.e. with $d_i(\mathbf{x}) = t_i$, are fully deformed because for them $b(\mathbf{x}) = 1$ holds. Vertices on the outer boundary, i.e. with $d_o(\mathbf{x}) = t_o$, are not deformed because for them $b(\mathbf{x}) = 0$ holds. For all other vertices, the deformation is smoothly blended between full and zero deformation.

3.3 Piecewise Deformation

By simply applying such a deformation to the whole mesh, vertices outside of the support region will be deformed as well and vertices in the handle region won't undergo a constant deformation, in general. We therefore need to perform the deformation in a piecewise fashion, which is quite simple. Vertices belonging to the handle region and the inner boundary are deformed in full, i.e. they undergo

a constant translation or rotation (e.g. the head and the boundary on the neck in Figure 2). Vertices in the support region (body between the boundaries in Figure 2), not belonging to any boundary, are deformed using the blended fields (4), (5). All other vertices are not deformed at all. Thanks to the two vanishing derives of the blending function at the boundaries, this piecewise deformation is C^1 continuous. Furthermore, the property of volume preservation still holds as long as no self-intersections occur. Self-intersections can only occur between the deforming parts of the mesh and the non-deforming parts.

Mathematically, we also could instead define the blending function in a piecewise fashion, such that the resulting deformation would be exactly the same. Technically, deforming the mesh in the piecewise manner described above is more efficent because the handle region can be deformed directly using a rigid transformation and the zero-deformation vertices are not considered at all.

3.4 Integration in Space Time

The description of the blending function is based on a time-independent context. However, since the inner boundary is actually moving over time, the blending function has to be updated within each integration step. This is straightforward: at the beginning of the integration, the inner boundary polygon is at its original position. Then, with each integration step, the position is updated by the amount corresponding to the step size. For instance, for a rotation, the polygon points are rotated step by step until the full rotation is reached. In [1], a more detailed description of the integration process can be found.

4 Applications

In principle, every deformation that is constructable as the cross product of two gradients can be used with the described approach. However, we confine ourselves to two simple, yet effective transformations: rotation and translation. Obviously, a scaling transformation wouldn't make sense, since we want to preserve the shape volume. As we will see in this section, this toolset allows for a variety of useful deformations.

In order to control translation and rotation, the user can place a *knob* somewhere on the mesh surface, and a *joint* somewhere in space. In most of the figures in this paper, the knob is depicted as a yellow stick, e.g. on top of the bust in Figure 2. The joint is a small white sphere, usually placed somewhere in the support region. The knob resembles typical *Gizmo* objects found in many shape modeling systems, which can be used to control transformations by grabbing and dragging it at different points. The knob is a simplified version because only translation and rotation are supported.

4.1 Rotation

In order to rotate the handle region, the user drags the knob, where the knob movement is constrained to a fixed radius about the joint position. From the displacement of the knob position, the rotation axis and angle can be determined

Fig. 4. Popular application scenarios for boundary constraint modeling

with respect to the joint position. Using the joint position as rotation center **c** in (3), a deformation that bends the shape can be accomplished by integrating the mesh vertices until the rotation angle is reached. In contrast to [1], where the shape is continually updated, the integration restarts from the original mesh every time the knob position changes. Figure 4, as well as Figure 2, shows this approach applied in various scenarios known from the Literature. The deformation looks rather realistic thanks to the volume-preservation and avoidance of self-intersections and even high resolution models can be deformed interactively.

Figure 5 demonstrates how local details are deformed: the "teeth" of the comb-like shape don't touch each other during deformation and their distortion seems appropriate with respect to the global deformation.

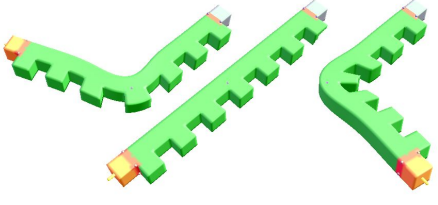


Fig. 5. Local details are slightly distorted in strongly deformed areas (left) and never intersect with each other (right)

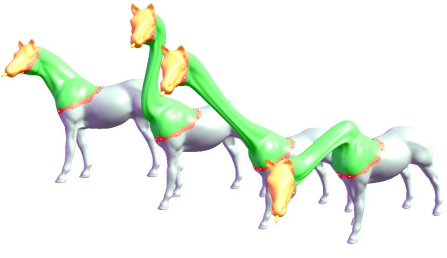


Fig. 6. By translating the horse head, the neck deforms in a natural manner

Also twisting is possible by simply using the vector between joint and knob as rotation axis. A more uniform twisting deformation can be achieved with two quadratic scalar fields [1].

4.2 Translation

When the user wants to translate the handle region, he or she can drag the knob freely in space. The joint is ignored for this deformation type. However, also the translation (2) requires a center point **c**. In this case we use the barycenter of the control points of the inner boundary.

In Figure 6, the user drags the head of a horse model. The shape of the neck automatically adapts to the new position. Due to the constant volume, the neck becomes thinner when the head is pulled.

Interesting effects can be achieved by carefully selecting boundaries and moving the handle parallel to the surface: as shown in Figure 7, the deformation automatically produces "wrinkles" on the cheek of the face, which is a result of the volume preservation and the prevention of foldovers. Although the "wrinkles" appear to be rather strong, no self-intersections of the surface occur.

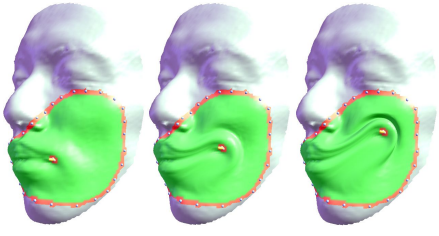


Fig. 7. "Wrinkles" can be produced by translating the handle accross the surface. Although they appear to be rather strong, no self-intersections of the surface occur.

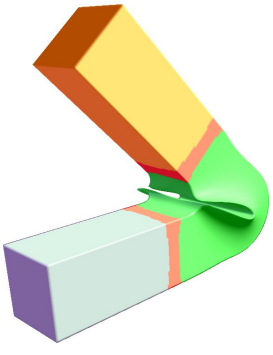


Fig. 8. When the boundaries move close to each other, the shape is distorted

5 Implementation and Performance

As shown in [1], the performance of the integration can be increased by a large amount by shifting the computation to the GPU, where multiple path lines can be integrated in parallel. This is also possible with our approach: since the number of boundary control points is usually low, these points can be passed to the shader as a simple array. We implemented two vertex programs, one for translation and one for rotation, which can be controlled by passing translation vector, rotation axis, angle etc. to them. During the integration, the polygon points of the inner boundary are updated internally with respect to the translation/rotation, as described in Section 3.4. Except for the extraction of the handle and support region (Section 3.3), no further preprocessing is required. After integration, vertex positions are read back from video memory and the mesh normals are computed. An alternative approach would be to compute normals directly on the GPU using the Jacobian of the vector field, similar to [18]. This would decrease the integration performance because of the necassary computation of Jacobians in each integration step, but would clear the CPU from doing this task and redundatize the readback of vertex positions. However, we

have not tested this alternative yet. As we will see in the following performance analysis, the normal computation on the CPU makes only a small fraction of the total deformation time.

The performance of the approach strongly depends on the "amount" of deformation, i.e. how far the handle is translated or rotated. This is because the numerical path line integration adapts its step size according to the complexity of the vector field and the duration of the integration. In order to present a meaningful statement about performance, we measured the times of usual "real-world" deformations, namely the ones depicted in Figure 4. Table 1 lists the deformed shapes (from left to right in Figure 4) and the benchmark results. *v/s*

Table 1. Performance benchmark: complex meshes can be deformed interactively

shape	vertices	boundary points	*v/s* (integration)	*v/s* (integration + normals)
box	47,296	8	788,267	647,890
dragon	86,814	23	413,400	369,421
leg 1	31,014	17	449,478	382,889
leg 2	31,014	17	443,057	364,871
finger	2725	14	454,167	454,167

(integration) is the number of vertices per second for integration only and *v/s (integration + normals)* the number of vertices per second for integration plus normal computation. The measurements were made on a 2.6 GHz Opteron CPU and a GeForce 6800 GT graphics card. They show that complex meshes can be deformed interactively.

6 Conclusion

We presented a shape deformation technique based on vector field integration which incorporates implicit boundaries to steer the impact of the deformation.

By using the VFSD technique [1], our deformations are volume-preserving and foldover-free, giving the user the impression of working with real, incompressible material. While the original VFSD approach defined the regions of influence by simple implicit objects, our method constructs a smooth blending function based on implicit boundaries. That way, the user can specifiy the impact of the deformation directly on the surface of the shape.

Thanks to the polygonal representation of the implicit boundaries, they are independent of the resolution of the deformed mesh. In most cases, a small number of control points suffices to define the boundaries.

Since the description of boundaries is simple (a small set of points), the numerical path line integration can be computed on the GPU and even complex models can be deformed interactively.

There are some restrictions of our approach which should be addressed in future research.

Self-intersections. Self-intersections are only avoided for the deforming regions of the shape surface because the deformation is carried out in a piecewise fashion. It is e.g. possible to bend the finger in Figure 4 such that the finger tip intersects the thumb or other parts of the hand. An additional collision detection would solve the problem, but would also drop performance.

Close boundaries. When inner and outer boundary are close to each other, the gradient of the resulting blending function is high in these regions. This can result in unpleasing deformations. E.g., when an extreme bending is performed (Figure 8), the boundaries approach each other, and the box is distorted more and more at its center (but nervertheless preserves its volume). A possible solution would be to perform such deformations (even more) piecewise, by using for example a third boundary between inner and outer boundary and constructing two blending functions: the first blends between inner and central boundary, the other between central and outer boundary.

References

1. von Funck, W., Theisel, H., Seidel, H.P.: Vector field based shape deformations. ACM Trans. Graph. 25(3), 1118–1125 (2006)
2. Welch, W., Witkin, A.: Variational surface modeling. In: SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pp. 157–166. ACM Press, New York (1992)
3. Taubin, G.: A signal processing approach to fair surface design. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp. 351–358. ACM Press, New York (1995)
4. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 105–114. ACM Press, New York (1998)
5. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. ACM Trans. Graph. 23(3), 630–634 (2004)
6. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: Proceedings of Shape Modeling International, pp. 181–190. IEEE Computer Society Press, Los Alamitos (2004)
7. Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, Eurographics Association, pp. 179–188 (2004)
8. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.Y.: Mesh editing with poisson-based gradient field manipulation. ACM Trans. Graph. 23(3), 644–651 (2004)
9. Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D.: Linear rotation-invariant coordinates for meshes. ACM Trans. Graph. 24(3), 479–487 (2005)
10. Zayer, R., Rössl, C., Karni, Z., Seidel, H.P.: Harmonic guidance for surface deformation. In: Computer Graphics Forum, Proceedings of Eurographics 2005, Dublin, Ireland, Eurographics, vol. 24, pp. 601–609. Blackwell, Oxford (2005)

11. Sederberg, T., Parry, S.: Free-form deformation of solid geometric models. In: SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp. 151–160. ACM Press, New York (1986)
12. Coquillart, S.: Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In: SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques, pp. 187–196. ACM Press, New York (1990)
13. MacCracken, R., Joy, K.: Free-form deformations with lattices of arbitrary topology. In: SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 181–188. ACM Press, New York (1996)
14. Barr, A.: Global and local deformations of solid primitives. In: SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pp. 21–30. ACM Press, New York (1984)
15. Singh, K., Fiume, E.: Wires: a geometric deformation technique. In: SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 405–414. ACM Press, New York (1998)
16. Hirota, G., Maheshwari, R., Lin, M.: Fast volume-preserving free form deformation using multi-level optimization. In: Proceedings Solid Modeling and applications, pp. 234–245 (1992)
17. Hsu, W., Hughes, J., Kaufman, H.: Direct manipulation of free-form deformations. In: SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pp. 177–184. ACM Press, New York (1992)
18. Botsch, M., Kobbelt, L.: Real-time shape editing using radial basis functions. Computer Graphics Forum (Proceedings Eurographics 2005) 24(3), 611–621 (2005)
19. Desbrun, M., Gascuel, M.P.: Animating soft substances with implicit surfaces. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp. 287–290. ACM Press, New York (1995)
20. Rappoport, A., Sheffer, A., Bercovier, M.: Volume-preserving free-form solids. IEEE Transactions on Visualization and Computer Graphics 2(1), 19–27 (1996)
21. Aubert, F., Bechmann, D.: Volume-preserving space deformation. Comput. and Graphics 21(5), 6125–6639 (1997)
22. Angelidis, A., Cani, M.P., Wyvill, G., King, S.: Swirling-sweepers: Constant-volume modeling. In: Computer Graphics and Applications, 12th Pacific Conference on (PG'04), pp. 10–15 (2004)
23. Mason, D., Wyvill, G.: Blendeforming: Ray traceable localized foldover-free space deformation. In: CGI '01: Proceedings of the International Conference on Computer Graphics, Washington, DC, USA, p. 183. IEEE Computer Society, Los Alamitos (2001)
24. Gain, J.E., Dodgson, N.A.: Preventing self-intersection under free-form deformation. IEEE Transactions on Visualization and Computer Graphics 7(4), 289–298 (2001)
25. Angelidis, A., Wyvill, G., Cani, M.P.: Sweepers: Swept user-defined tools for modeling by deformation. In: Proceedings of Shape Modeling and Applications, pp. 63–73. IEEE, Orlando (June 2004)
26. Biswas, A., Shapiro, V.: Approximate distance fields with non-vanishing gradients. Graphical Models 66(3), 133–159 (2004)
27. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM national conference, pp. 517–524. ACM Press, New York (1968)

Tuning Subdivision Algorithms Using Constrained Energy Optimization

Ingo Ginkel and Georg Umlauf

Geometric Algorithms Group, Department of Computer Science,
University of Kaiserslautern, Germany
{ginkel,umlauf}@informatik.uni-kl.de
http://www-umlauf.informatik.uni-kl.de

Abstract. In this paper a method is presented to fair the limit surface of a subdivision algorithm around an extraordinary point. The eigenvalues and eigenvectors of the subdivision matrix determine the continuity and shape of the limit surface. The dominant, sub-dominant and subsub-dominant eigenvalues should satisfy linear and quadratic equality- and inequality-constraints to guarantee continuous normal and bounded curvature globally. The remaining eigenvalues need only satisfy linear inequality-constraints. In general, except for the dominant eigenvalue, all eigenvalues can be used to optimize the shape of the limit surface with our method.

1 Introduction

Subdivision algorithms are a well established tool in computer graphics to model objects with free-form geometry. Nevertheless, the shape of the resulting subdivision surfaces have artifacts [10,24]. This is one reason why subdivision surfaces are not widely used in CAD applications. Hence, there are various approaches to tune the subdivision algorithms to reduce these artifacts. Many approaches were taken that modify the differential geometric properties but not the fairness of the resulting subdivision surfaces [3,6,8,12,13,20,21,23].

To further improve the fairness of the surfaces optimization techniques have been used to construct optimized stationary schemes. In [2] also the eigenvectors are modified by computing the stencils in an optimization process. The resulting subdivision algorithms approximate the desired eigenvector behavior. Based on the analysis in [10,16] a different approach is presented in [1]. Here, the eigenvalues are determined such that the necessary conditions for bounded Gauss curvature are satisfied and the variation of curvature of the central surfaces in the shape chart is minimized. This is a non-linear optimization process which can only be approximated. It leads to subdivision algorithms with an eigenvalue distribution that approximately minimizes variation of Gauss curvature and the probability for a so-called hybrid shape. Nevertheless, in both approaches the immediate influence of the optimization on the fairness of the subdivision surface cannot be controlled directly.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 166–176, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The approaches taken in [7,11] aim at fairing the limit surface. Instead to tune the respective subdivision algorithms they integrate optimization to the subdivision rules. The points of the fine control nets are computed such that a local or global energy functional is minimized. For both approaches this requires the solution of a global system of linear equations.

The principle of the tuning method presented here is based on a diagonalization of the subdivision matrix S. The eigenvalues of any stationary, linear and symmetric subdivision algorithm can be changed to satisfy the necessary conditions for bounded Gauss curvature using the technique described in [21]. This can be generalized to calculate the eigenvalue changes in an optimization process incorporating the necessary conditions as constraints imposing equality- and inequality-constraints on the dominant, sub-dominant and subsub-dominant eigenvalues of the subdivision matrix. The remaining eigenvalues must stay within certain intervals. Because the stencils depend linearly on the eigenvalues and a quadratic energy functional like the thin-plate energy depends quadratically on the control points, stencils can be computed that generate surfaces minimizing the respective functional. This means the scheme is no longer stationary, but it does not require the solution of a global system of equations.

The rest of the paper is structured into three sections. First the general concept of energy minimization for tuning subdivision surfaces is presented (Section 2). In Section 3 the constraints for the optimization process are derived from the Fourier analysis of subdivision schemes. These constraints include regularity, bounded curvature and changes that only affect the local neighborhood of the extraordinary point. The results of this method used to tune the algorithm of Catmull-Clark are shown in Section 4.

2 Energy Minimization Concept

A linear subdivision algorithm computes from a coarse initial net of control points $\mathcal{N}_0$ a refined net of control points $\mathcal{N}_1$ by taking finite affine combinations of control points of $\mathcal{N}_0$ to compute the control points of $\mathcal{N}_1$. This process is iterated to generate a sequence of control nets $(\mathcal{N}_r)_{r\geq 0}$ which converges to the subdivision surface.

For a primal subdivision algorithm the subdivision matrix S maps the k-ring neighborhood $\mathcal{C}_r, r \geq 0$, of an n-valent vertex $\mathbf{c}_r$ in $\mathcal{N}_r$ net to the k-ring neighborhood $\mathcal{C}_{r+1}$ of an n-valent vertex $\mathbf{c}_{r+1}$ in $\mathcal{N}_{r+1}$. Note that S is a square matrix and k depends on the size of the stencils. Each control net $\mathcal{C}_r$ defines a surface ring $\mathbf{x}_r$ which is represented as a linear combination of bi-variate, real valued, piecewise smooth functions $\mathbf{f} = [f_1, \ldots, f_l]$ forming a partition of unity corresponding to the column vector of control points C_r of $\mathcal{C}_r$

$$\mathbf{x}_r = \mathbf{f} \cdot C_r, \qquad r \geq 0.$$

Example surface rings are shown in Figure 1 for the algorithm of Catmull-Clark.

The goal of this paper is to construct a sequence of spline rings by a modified subdivision matrix $\widetilde{S}$, where each spline ring $\mathbf{x}_r$ has minimal energy $F(\mathbf{x}_r)$.

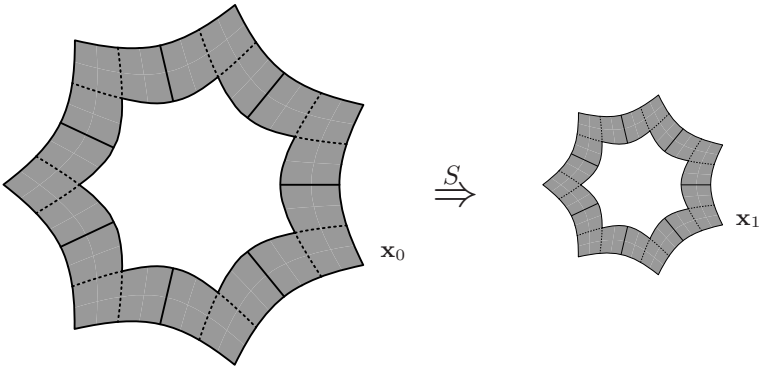


Fig. 1. Two consecutive surface rings $\mathbf{x}_0$ and $\mathbf{x}_1$ for the algorithm of Catmull-Clark at a 7-valent vertex

If we take an energy functional like the thin-plate energy of a spline surface $\mathbf{x} : \mathbb{R}^2 \to \mathbb{R}^3$ with control points $\mathbf{p}_1, \ldots, \mathbf{p}_L \in \mathbb{R}^3$, it can be written as quadratic form, see e.g. [9]

$$F(\mathbf{x}) = \sum_{c=1}^{3} [\mathbf{p}_1^c, \ldots, \mathbf{p}_L^c] \cdot E \cdot [\mathbf{p}_1^c, \ldots, \mathbf{p}_L^c]^t,$$

where $\mathbf{p}_j^c$ denotes the c-th coordinate of control point $\mathbf{p}_j$. The matrix E contains the energies of the spline basis functions. If $\mathbf{f}$ consists of b-spline or box-spline functions, the surface rings $\mathbf{x}_r$ consist of s polynomial pieces as shown schematically in Figure 1 with $s = 3 \cdot 7$. The energy $F(\mathbf{x}_r)$ can then be computed as

$$F(\mathbf{x}_r) = \sum_{c=1}^{3} \sum_{j=1}^{s} (C_r^c)^t P_j^t E P_j C_r^c = \sum_{c=1}^{3} (C_r^c)^t \cdot \underbrace{\left(\sum_{j=1}^{s} P_j^t E P_j \right)}_{=: G} \cdot C_r^c,$$

where C_r^c denotes the column vector of c-th coordinates of the control points of C_r. The matrices P_j select those points from C_r that define the j-th patch of $\mathbf{x}_r$. Thus, the energy of $\mathbf{x}_{r+1}$ is given by

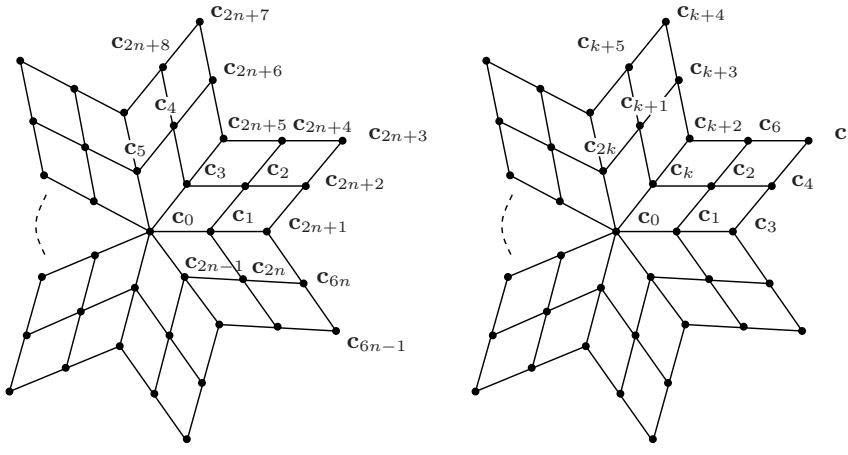
$$F(\mathbf{x}_{r+1}) = \sum_{c=1}^{3} (C_r^c)^t \widetilde{S}^t G \widetilde{S} C_r^c.$$

Since the optimization should be local, only a few stencils are modified, for example only the stencils for the extraordinary point and the control points in its one-ring neighborhood. If the control points are labeled circular around the

extraordinary vertex as shown for the algorithm of Catmull-Clark in Figure 2(a), the modified subdivision matrix $\widetilde{S}$ has the following structure

$$\widetilde{S} = \begin{bmatrix} \widetilde{S}_1 & 0 \\ S_2 & S_3 \end{bmatrix},$$

where $\widetilde{S}_1$ is the sub-matrix which contains the stencils that will be modified.



(a) Circular labeling of the control points

(b) Segmentwise labeling of the control points, where k is the number of points per segment

Fig. 2. Labeling of the control points around a vertex of valence n

This structure is due to the fact that the stencils for the new positions of the extraordinary vertex and the one-ring are local. For the standard algorithms like the algorithms of Catmull-Clark or Loop S_1 is square. Thus, the energy $F(\mathbf{x}_r)$ can be written as

$$F(\mathbf{x}_{r+1}) = \sum_{c=1}^{3} (C_r^c)^t \Bigg(\underbrace{\begin{bmatrix} \widetilde{S}_1 & 0 \\ 0 & 0 \end{bmatrix}}_{=:A} + \underbrace{\begin{bmatrix} 0 & 0 \\ S_2 & S_3 \end{bmatrix}}_{=:B} \Bigg)^t G \left(\begin{bmatrix} \widetilde{S}_1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ S_2 & S_3 \end{bmatrix} \right) C_r^c.$$

If $\widetilde{S}_1$ is diagonalizable, the eigenvalues of $\widetilde{S}_1$ can be used for optimization without affecting S_2 and S_3. To prepare the equations for the optimization we take a closer look at the matrix A, which contains the stencils that will be modified

$$AC_r^c = \begin{bmatrix} \widetilde{S}_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} C_{1,r}^c \\ C_{2,r}^c \end{bmatrix} = \begin{bmatrix} \widetilde{S}_1 C_{1,r}^c \\ 0 \end{bmatrix}.$$

Denote by R the Matrix of eigenvectors of $\widetilde{S}_1$, by L its inverse with rows $\mathbf{w}_1, \ldots, \mathbf{w}_{l_1}$ and by D the diagonal matrix of the corresponding eigenvalues $\lambda_1, \ldots, \lambda_{l_1}$ of $\widetilde{S}_1$. Then,

$$\begin{aligned}\widetilde{S}_1 C^c_{1,r} &= RDLC^c_{1,r} \\ &= R \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{l_1} \end{bmatrix} \begin{bmatrix} \mathbf{d}^c_1 \\ \vdots \\ \mathbf{d}^c_{l_1} \end{bmatrix} = R \begin{bmatrix} \mathbf{d}^c_1 & & \\ & \ddots & \\ & & \mathbf{d}^c_{l_1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{l_1} \end{bmatrix} =: M^c_1 \boldsymbol{\lambda}_1,\end{aligned}$$

where $\mathbf{d}^c_i$ is the c-th coordinate of the so-called eigencoefficient $\mathbf{d}_i = \mathbf{w}_i C^c_{1,r}$, and

$$AC^c_r = \begin{bmatrix} M^c_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ 0 \end{bmatrix} =: M^c \boldsymbol{\lambda}.$$

Keeping R and L fixed, the energy $F(\mathbf{x}_{r+1})$ is quadratic in λ_i, since M^c and B do not depend on $\boldsymbol{\lambda}$, i.e.

$$\begin{aligned}F(\mathbf{x}_{r+1}) &= \sum_{c=1}^{3} (C^c_r)^t (A+B)^t G (A+B) C^c_r \\ &= \sum_{c=1}^{3} \left(\boldsymbol{\lambda}^t (M^c)^t G M^c \boldsymbol{\lambda} + 2 (C^c_r)^t B^t G M^c \boldsymbol{\lambda} + (C^c_r)^t B^t G B C^c_r \right). \quad (1)\end{aligned}$$

Using this representation of $F(\mathbf{x}_{r+1})$ the eigenvalues $\boldsymbol{\lambda}$ can be used for optimization with a prescribed number of stencils that will be modified.

3 Constrained Energy Minimization

To make the algorithm well-behaved we need certain constraints such as affine invariance or constraints that guarantee C^1 regularity of the limit surface. Many of these constraints become visible in the discrete Fourier analysis of the subdivision matrix.

Assume that the subdivision matrix S has only real eigenvalues $\lambda_1, \ldots, \lambda_l$ ordered by modulus

$$|\lambda_1| \geq \cdots \geq |\lambda_l| \geq 0$$

corresponding to the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_l$, i.e. $S\mathbf{v}_i = \lambda_i \mathbf{v}_i$ for $i = 1, \ldots, l$. If the subdivision algorithm uses only affine combinations, the dominant eigenvalue λ_1 is one and simple [19], i.e.

$$\lambda_1 = 1, \quad (2)$$

$$\lambda_1 > |\lambda_2|. \quad (3)$$

Then, the sequence of surface rings $(\mathbf{x}_r)_{r \geq 0}$ converges to the so-called extraordinary point $\mathbf{s}$ on the subdivision surface corresponding to $\mathbf{c}_0$.

If the subdivision algorithm is rotationally symmetric, the subdivision matrix S is block-circulant and similar to a block-diagonal matrix $\widehat{S}$ using the labeling of control points as in Figure 2(a)

$$S = \mathrm{circ}(S^0, \ldots, S^{n-1}) \sim \widehat{S} = \mathrm{diag}(\widehat{S}^0, \ldots, \widehat{S}^{n-1})$$

with the same eigenvalues, i.e. $\mathrm{spec}(S) = \bigcup_i \mathrm{spec}(\widehat{S}^i)$. Because $\widehat{S}$ is computed by a discrete Fourier transform as

$$\widehat{S}^i = \sum_{\ell=0}^{n-1} \omega_n^{-i\ell} S^\ell, \qquad \omega_n := \exp(2\pi\sqrt{-1}/n), i = 0, \ldots, n-1,$$

an eigenvalue ν of S is said to have Fourier index $\mathcal{F}(\nu)$ if $\nu \in \mathrm{spec}(\widehat{S}^{\mathcal{F}(\nu)})$. For subdivision surface to be regular and normal-continuous with bounded Gauss curvature of arbitrary sign at $\mathbf{s}$ the following conditions are sufficient [22,17]:

(i) The sub-dominant eigenvalue $\lambda_2 =: \lambda$ is a double, real eigenvalue and $\mathbf{v}_2, \mathbf{v}_3$ are linearly independent, i.e.

$$\begin{aligned} \lambda_2 &= \lambda_3, \\ |\lambda_2| &> |\lambda_4|, \ldots, |\lambda_l|. \end{aligned} \tag{4}$$

(ii) The so-called characteristic map $\mathbf{f} \cdot [\mathbf{v}_2 \mathbf{v}_3]$ is regular and injective.

(iii) The subsub-dominant eigenvalue $\lambda_4 =: \mu$ is real and satisfies

$$\mu = \lambda^2. \tag{5}$$

(iv) The subsub-dominant eigenvalue μ is a triple eigenvalue and $\mathbf{v}_4, \mathbf{v}_5, \mathbf{v}_6$ are linearly independent, i.e.

$$\lambda_4 = \lambda_5 = \lambda_6, \tag{6}$$

$$\lambda_4 > |\lambda_7|, \ldots, |\lambda_l|. \tag{7}$$

(v) The subsub-dominant eigenvalue μ has Fourier indices $0, 2$ and $n-2$.

The constraints (2)–(6) impose quadratic and linear equality- and inequality-constraints on the eigenvalues $\lambda_1, \ldots, \lambda_6$. The remaining eigenvalues $\lambda_i, i \geq 7$, can be chosen arbitrarily as long as (7) and Conditions (ii) and (v) are not violated. Combining the constraints with Equation (1) a quadratic equation with quadratic equality-constraints and linear inequality- and equality-constraints must be solved. Note that leaving the sub-dominant eigenvalues unchanged, the quadratic equality constraints disappear leading to a quadratic program.

4 Constrained Energy Minimization for the Algorithm of Catmull-Clark

Using the labeling of control points as in Figure 2(b), the Fourier blocks $\widehat{S}^i, i = 0, \ldots, n-1$, of S of the algorithm of Catmull-Clark are given by

$$\widehat{S^i} = \left[\begin{array}{c|ccc|cccc|ccccccc}
\alpha\delta_{i,0} & \beta\delta_{i,0} & & \gamma\delta_{i,0} & & & & & & & & & & & \\
\frac{3}{8}\delta_{i,0} & \frac{3}{8}+\frac{1}{8}c_i & & \frac{1}{16}+\frac{1}{16}\omega^{-i} & & & & & & & & & & & \\
\frac{1}{4}\delta_{i,0} & \frac{1}{4}+\frac{1}{4}\omega^{i} & & \frac{1}{4} & & & & & & & & & & & \\
\hline
\frac{3}{32}\delta_{i,0} & \frac{9}{16}+\frac{1}{32}c_i & \frac{3}{32}+\frac{3}{32}\omega^{-i} & & \frac{3}{32} & \frac{1}{64} & 0 & \frac{1}{64}\omega^{-i} & & & & & & & \\
\frac{1}{16}\delta_{i,0} & \frac{3}{8}+\frac{1}{16}\omega^{i} & & \frac{3}{8} & \frac{1}{16} & \frac{1}{16} & 0 & 0 & & & & & & & \\
\frac{1}{64}\delta_{i,0} & \frac{3}{32}+\frac{3}{32}\omega^{i} & & \frac{9}{16} & \frac{1}{64}+\frac{1}{64}\omega^{i} & \frac{3}{32} & \frac{1}{64} & \frac{3}{32} & & & & & & & \\
\frac{1}{16}\delta_{i,0} & \frac{1}{16}+\frac{3}{8}\omega^{i} & & \frac{3}{8} & \frac{1}{16}\omega^{i} & 0 & 0 & \frac{1}{16} & & & & & & & \\
\hline
0 & \frac{3}{8} & \frac{1}{16}+\frac{1}{16}\omega^{-i} & & \frac{3}{8} & \frac{1}{16} & 0 & \frac{1}{16}\omega^{-i} & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & \frac{1}{4} & & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & \frac{1}{16} & & \frac{3}{8} & \frac{1}{16} & \frac{3}{8} & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & \frac{1}{16}\omega^{i} & & \frac{3}{8} & \frac{1}{16}\omega^{i} & \frac{1}{16} & \frac{1}{16} & \frac{3}{8} & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & \frac{1}{4}\omega^{i} & & \frac{1}{4} & \frac{1}{4}\omega^{i} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 &
\end{array}\right],$$

where $\omega := \exp(2\pi\sqrt{-1}/n)$, $c_i := \cos(2\pi i/n)$ and the Kronecker symbol $\delta_{i,0}$. From $\widehat{S^i}$ the eigenvalues of S can be computed as:

- the eigenvalue $\lambda_0 = 1$ from $\widehat{S}_0$,
- the eigenvalues $\mu_\alpha^\pm := (4\alpha - 1 \pm \sqrt{(4\alpha-1)^2 + 8\beta - 4})/8$ from $\widehat{S}_0$,
- the eigenvalues $\mu_i^\pm := (5 + c_i \pm \sqrt{(c_i+9)(c_i+1)})/16$ from $\widehat{S}_i, i = 1, \ldots, n-1$,
- the n-fold eigenvalues $\frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}$ from $\widehat{S}_i, i = 0, \ldots, n-1$,
- the $(7n-1)$-fold eigenvalue 0 from $\widehat{S}_i, i = 0, \ldots, n-1$.

The upper left 3×3 block of $\widehat{S}_i$ corresponds to the extraordinary vertex and its one-ring neighborhood. Diagonalizing this block, its eigenvalues can be used for optimization subject to the following constraints:

$$\lambda_0 = 1, \tag{8}$$

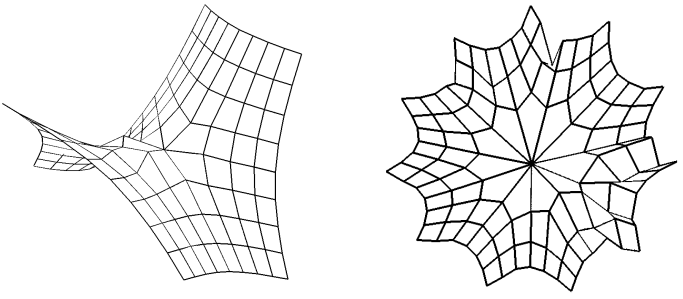
$$1/\sqrt{8} < \mu_1^+ < 1, \tag{9}$$

$$\mu_\alpha^+ = \mu_2^+ = (\mu_1^+)^2, \tag{10}$$

$$\mu_i^+ = \mu_{n-i}^+ \text{ and } \mu_i^- = \mu_{n-i}^- \text{ for } i = 1, \ldots, \lfloor n/2 \rfloor, \tag{11}$$

$$|\mu_\alpha^-|, |\mu_i^\pm|, |\mu_1^-|, |\mu_2^-| < \mu_2^+ \text{ for } i = 3, \ldots, \lfloor n/2 \rfloor. \tag{12}$$

The results of an optimization with these contraints using the libraray [14] are shown in Figures 5 . The control nets shown in Figure 3 contain irregular vertices of valence 7 and 11. It does not seem necessary to hold up the triple subsubdominant eigenvalue, since hybrid shapes can still occur (see Figure 5 bottom left). Instead we replace the constraints $\mu_2^+ = (\mu_1^+)^2$ by $|\mu_2^+| < (\mu_1^+)^2$ for elliptic and $\mu_\alpha^+ = (\mu_1^+)^2$ by $|\mu_\alpha^+| < (\mu_1^+)^2$ for hyperbolic shapes. We leave the decision whether the shape is going to be elliptic or hyperbolic to the optimization by calculating both variants and taking the one that produces lower energy for the spline rings, see Figure 4 and Figure 5 bottom right.



(a) Control net with an irregular vertex of valence 7.

(b) Control net with an irregular vertex of valence 11.

Fig. 3. Two different control nets for the algorithm of Catmull-Clark

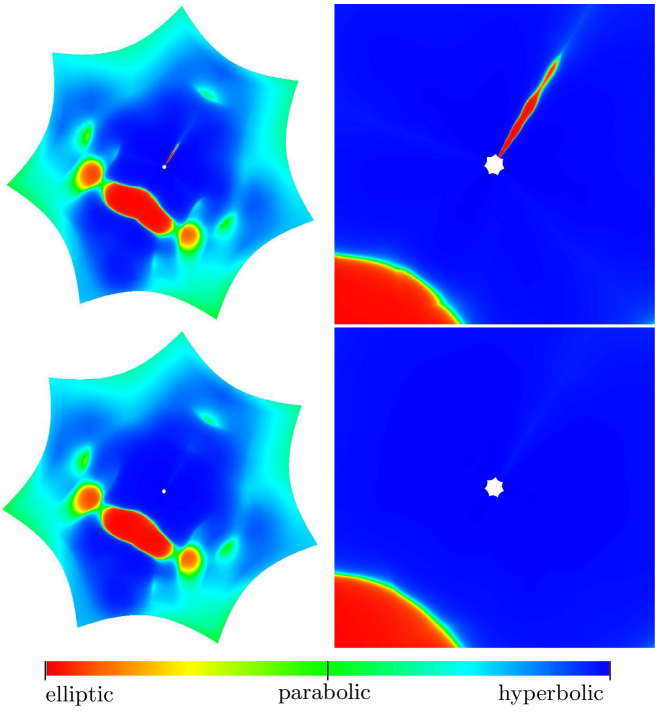


Fig. 4. Comparison of two different subdivision surfaces corresponding to the control net in Figure 3(a) after 8 subdivision steps. *Top*: Modified Catmull-Clark algorithm ([20]) with bounded curvature of arbitrary sign. *Bottom*: Optimized Catmull-Clark algorithm with double subsub-dominant eigenvalue for hyperbolic shape. The right column shows the corresponding zoom at the extraordinary point. For color images please refer to our website.

Table 1. Optimized values for μ_α^+,μ_2^+, μ_{n-2}^+ and comparision of the energies of the innermost two spline rings at the corresponding subdivision level r with and without optimization for the algorithm of Catmull-Clark for the surfaces in Figure 4

r	μ_α^+	μ_2^+, μ_{n-2}^+	$F(\mathbf{x})$ without optimization	$F(\mathbf{x})$ with optimization
1	0.365611	0.365768	28.08448	25.96919
2	0.354541	0.355186	7.112990	6.828013
3	0.347269	0.350177	2.246211	2.192127
4	0.345149	0.348187	0.751045	0.736397
5	0.341162	0.347430	0.256974	0.252283
6	0.337389	0.347161	0.088712	0.087122
7	0.324970	0.347065	0.030727	0.030179
8	0.203355	0.347033	0.010656	0.010466

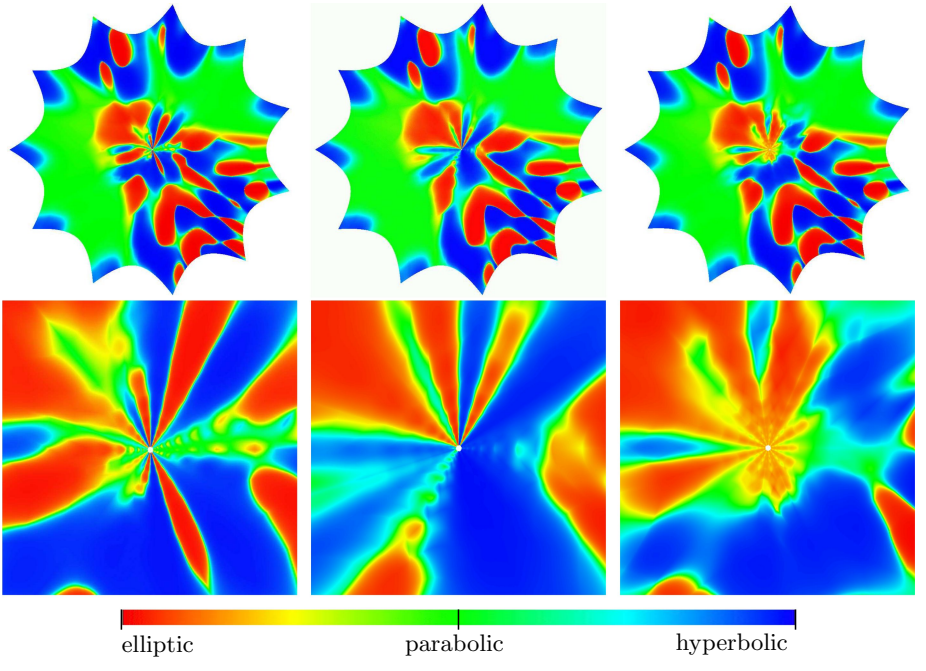


Fig. 5. Comparison of three different subdivision surfaces corresponding to the control net in Figure 3(b) after 12 subdivision steps. *Left*: Modified Catmull-Clark algorithm ([20]) with bounded curvature of arbitrary sign. *Middle*: Optimized Catmull-Clark algorithm with triple subsub-dominant eigenvalue. *Right*: Optimized Catmull-Clark algorithm with single subsub-dominant eigenvalue for elliptic shape. The top row shows the surface and the bottom row shows the corresponding zoom at the extraordinary point. For color images please refer to our website.

The functional that has been used for the optimization is

$$F(\mathbf{x}) = \int_{[0,1]^2} \left(\mathbf{x}_{uuu}^2 + 3\mathbf{x}_{uuv}^2 + 3\mathbf{x}_{uvv}^2 + \mathbf{x}_{vvv}^2\right) du\, dv, \tag{13}$$

where $\mathbf{x}_u$ and $\mathbf{x}_v$ denote the partial derivatives of $\mathbf{x}$ with respect to u and v. We chose a parameterization dependent third order measure as a reasonable approximation to the variation of curvature. Since second order measures may lead to flat surfaces, minimizing the variation of curvature rather than its magnitude leads to fairer and more pleasing surfaces [15].

The influence of the optimization is visible in the visualization of the Gauss curvature. For the surface in Figure 4, we additionally document the tendency towards a hyperbolic configuration in Table 1. It shows the optimized values for μ_α^+, μ_2^+ and μ_{n-2}^+ for the first 8 subdivision steps and compares the energies of the innermost two spline rings at the corresponding subdivision level with and without optimization.

Remark 1. Changes in the stencils for the one-ring around the extraordinary vertex by the optimization influence the spline rings $\mathbf{x}_{r+1}$ and $\mathbf{x}_r$. Therefore, the optimization has to take both rings into account by choosing the size of C_r, S and G such that four rings of control points are used in the optimization.

Remark 2. Since the optimization changes μ_1^+, the proof for regularity and injectivity of the characteristic map is no longer valid. It needs to be re-done using the technique in [25] for the new choice of μ_1^+.

5 Conclusion

We have presented a technique to tune subdivision algorithms using constrained energy optimization. This allows to obtain bounded curvature for the limit behavior in combination with a fair shape around the extraordinary point.

As future work we will determine an interval of valid values for μ_1^+, for which the characteristic map is regular and injective. This interval would be included into the optimization as further constraint.

References

1. Augsdörfer, U., Dodgson, N.A., Sabin, M.A.: A New Way to Tune Subdivision. In: Desbrun, M., Pottmann, H. (eds.) Eurographics Symposium on Geometry Processing (2005)
2. Barthe, L., Kobbelt, L.: Subdivision Scheme Tuning Around Extraordinary Vertices. Comput. Aided Geom. Design 21, 561–583 (2004)
3. Catmull, E., Clark, J.: Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. Comput.-Aided Des. 10, 350–355 (1978)
4. Doo, D.W.H., Sabin, M.: Behaviour of Recursive Division Surfaces near Extraordinary Points. Comput.-Aided Des. 10, 356–360 (1978)
5. Ginkel, I., Umlauf, G.: Analyzing a Generalized Loop Subdivision Scheme. Computing (to appear 2006)

6. Ginkel, I., Umlauf, G.: Loop Subdivision with Curvature Control. In: Polthier, K., Sheffer, A. (eds.) Eurographics Symposium on Geometry Processing, pp. 163–171 (2006)
7. Halstead, M.A., Kass, M., DeRose, T.D.: Efficient, Fair Interpolation using Catmull-Clark Surfaces. SIGGRAPH '93, pp. 35–44 (1993)
8. Holt, F.: Towards a Curvature-Continuous Stationary Subdivision Algorithm Z. Angew. Math. Mech. 76, 423–424 (1996)
9. Kallay, M.: Constrained Optimization in Surface Design. In: Falcidieno, B., Kunii, T.L. (eds.) Modeling in Computer Graphics, pp. 85–93 (1993)
10. Karciauskas, K., Peters, J., Reif, U.: Shape Characterization of Subdivision Surfaces—Case Studies. Comput. Aided Geom. Design 21, 601–614 (2004)
11. Kobbelt, L.: A Variational Approach to Subdivision. Comput. Aided Geom. Design 13, 743–761 (1996)
12. Loop, C.: Smooth Subdivision Surfaces Based on Triangles Master's Thesis, University of Utah (1987)
13. Loop, C.: Bounded Curvature Triangle Mesh Subdivision with the Convex Hull Property. Visual Comput. 18, 316–325 (2002)
14. Meza, J.C., Oliva, R.A., Hough, P.D., Williams, P.J.: OPT++: An Object-Oriented Class Library for Nonlinear Optimization
15. Moreton, H., Séquin, C.: Minimum Variation Curves and Surfaces for Computer-Aided Geometric Design. In: Sapidis, N.S. (ed.) Designing Fair Curves and Surfaces, SIAM, Philadelphia, pp. 123–159 (1994)
16. Peters, J., Reif, U.: Shape Characterization of Subdivision Surfaces—Basic Principles. Comput. Aided Geom. Design 21, 585–599 (2004)
17. Peters, J., Umlauf, G.: Gaussian and Mean Curvature of Subdivision Surfaces. In: Cipolla, R., Martin, R. (eds.) The Mathematics of Surfaces IX, pp. 59–69. Springer, Heidelberg (2000)
18. Prautzsch, H.: Smoothness of Subdivision Surfaces at Extraordinary Points. Adv. Comput. Math. 9, 377–389 (1998)
19. Prautzsch, H., Boehm, W., Paluszny, M.: Bézier and B-Spline Techniques. Springer, Heidelberg (2002)
20. Prautzsch, H., Umlauf, G.: A G^2-subdivision Algorithm. Computing 13, 217–224 (1998)
21. Prautzsch, H., Umlauf, G.: A G^1 and G^2 Subdivision Scheme for Triangular Nets. Int. J. Shape Model. 6, 21–35 (2000)
22. Reif, U.: A Unified Approach to Subdivision Algorithms near Extraordinary Vertices. Comput. Aided Geom. Design 12, 153–174 (1995)
23. Sabin, M.A.: Cubic Recursive Division with Bounded Curvature. In: Laurent, P.J., Le Méhauté, A., Schumaker, L.L. (eds.) Curves and Surfaces, pp. 411–414. Academic Press, San Diego (1991)
24. Sabin, M.A., Barthe, L.: Artifacts in Recursive Subdivision Surfaces. In: Cohen, A., Merrien, J.-L., Schumaker, L.L. (eds.) Curve and Surface Fitting Saint-Malo 2002, pp. 353–362, Nashboro Press (2003)
25. Umlauf, G.: A Technique for Verifying the Smoothness of Subdivision Schemes. In: Lucian, M.L., Neamtu, M. (eds.) Geometric Modeling and Computing, pp. 513–521, Nashboro Press (2004)

Description of Surfaces in Parallel Coordinates by Linked Planar Regions

Chao-Kuei Hung[1] and Alfred Inselberg[2]

[1] CSIE Department, Shu-Te University, Kaoshiung, Taiwan
ckhung@mail.stu.edu.tw
[2] School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel
aiisreal@math.tau.ac.il

Abstract. An overview of the methodology covers the representation (i.e. visualization) of multidimensional lines, planes, flats, hyperplanes, and curves. Starting with the visualization of hypercubes of arbitrary dimension the representation of smooth surfaces is developed in terms of linked planar regions. The representation of *developable*, *ruled*, non-orientable, *convex* and *non-convex* surfaces in $\mathbb{R}^3$ with generalizations to $\mathbb{R}^N$ are presented enabling efficient visual detection of surface properties. The parallel coordinates methodology has been applied to collision avoidance algorithms for air traffic control (3 USA patents), computer vision (1 USA patent), data mining (1 USA patent), optimization and elsewhere.

1 Introduction

Do It in Parallel!
Parallel coordinates (abbreviated as ∥-coords) transforms N-dimensional objects into distinct planar patterns enabling our fantastic pattern-recognition, aided by interactivity, to obtain insight about multivariate problems. Starting with an overview of the fundamentals the presentation leads to the representation of surfaces in ∥-coords focusing on core ideas and avoiding unessential technicalities (for details see [1] and the forthcoming textbook [2]).

On the plane with xy-Cartesian coordinates a vertical line, labeled $\bar{X}_i$, is placed at each $x = i - 1$ for $i = 1, 2, ..., N$. These are the axes of the parallel coordinate system for $\mathbb{R}^N$. A point $C = (c_1, c_2, ..., c_N)$ is mapped into the polygonal line $\overline{C}$ whose N-vertices with xy-coords $(i - 1, c_i)$ are on the parallel axes. In $\overline{C}$ the full lines, not just the segments between the axes, are included as illustrated in Fig. 1(left) for $N = 5$. The *representation* of an object S in ∥-coords is denoted by $\bar{S}$.

2 Overview of Parallel Coordinates

2.1 Duality in the Plane

A point $P = (p_1, p_2)$ on the plane is represented by a line on the points $(0, p_1)$ and (d, p_2). As illustrated in Fig. 1 (right), points on a line

$$\ell : x_2 = mx_1 + b, \tag{1}$$

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 177–208, 2007.
© Springer-Verlag Berlin Heidelberg 2007

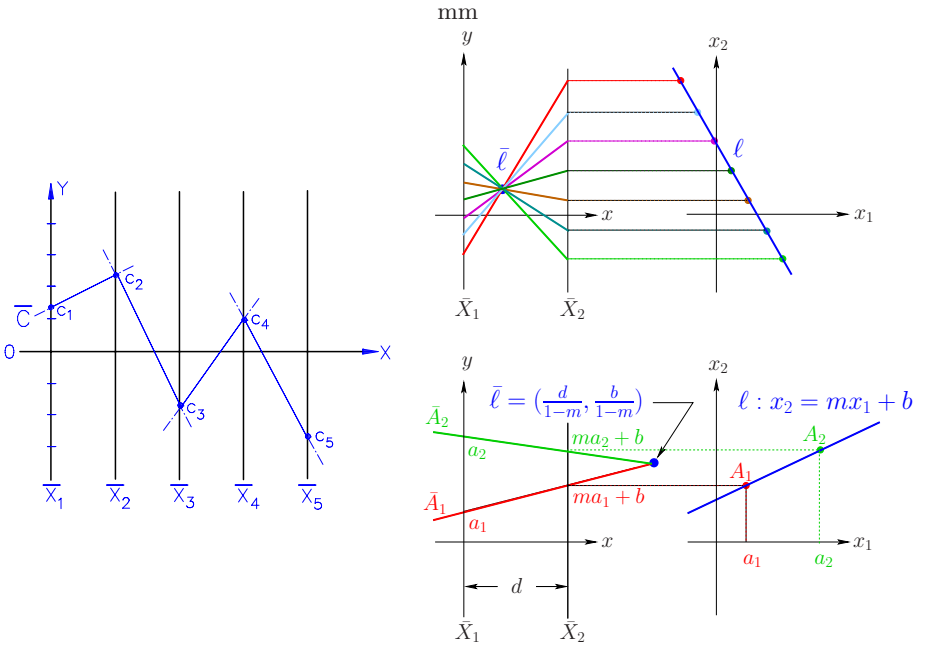


Fig. 1. (left) A point $C = (c_1, \ldots, c_5) \in \mathbb{R}^5$ is represented by a polygonal line $\bar{C}$. (right) Parallel coordinates induce a *point* $\leftrightarrow$ *line* duality in $\mathbb{R}^2$.

are represented in ∥-coords by lines intersecting at the point

$$\bar{\ell} : \left(\frac{d}{1-m}, \frac{b}{1-m} \right), \tag{2}$$

where d is the inter-axes distance. In 2-D then ∥-coords induce a point $\leftrightarrow$ line duality (i.e. mapping points into lines and vice versa). Dualities properly reside in $\mathbb{P}^2$; the Projective rather than $\mathbb{R}^2$ the Euclidean plane. Here this is hinted by the denominator in eq. 2. As $m \to 1$ the point $\bar{l} \to \infty$ in the *constant* direction with slope b/d. The full duality exists in $\mathbb{P}^2$ for lines with $m = 1$ corresponding to ideal points i.e. 'directions'. Note that parallel lines are represented by points having the same value of x. Using homogeneous coordinates[1] the mapping is a linear transformation:

$$\ell : [m, -1, b] \to \bar{\ell} : (d, b, 1-m) \tag{3}$$

where the brackets $[,]$ denote line coordinates. One does not need expertise in projective geometry to work with ∥-coords but awareness is advisable to avoid blunders.

[1] Denoted for a planar point by a triple between ().

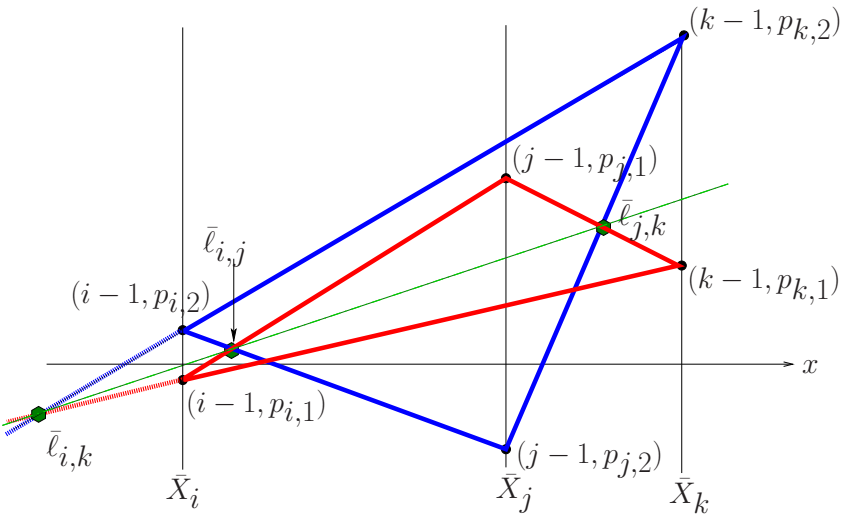


Fig. 2. The 3 points $\bar{\ell}_{i,j}, \bar{\ell}_{j,k}, \bar{\ell}_{i,k}$ are on a line $\bar{L}$ for distinct i, j, k

2.2 Lines in $\mathbb{R}^N$

A line ℓ in Euclidean N-space $\mathbb{R}^N$ is completely described in terms of $N-1$ independent projections on the $x_i x_j$ 2-planes given by

$$\ell_{i,j} : x_i = m_{i,j} x_j + b_{i,j} \ . \tag{4}$$

By the *line* $\rightarrow$ *point* mapping, eq. (2), these 2-D lines are represented by the points (assuming $i < j$)

$$\bar{\ell}_{i,j} : \left(\frac{j-i}{1-m_{i,j}} + (i-1), \frac{b_{j-i}}{1-m_{i,j}} \right) \tag{5}$$

where the distance between adjacent axes $\bar{X}_{i-1}$ and $\bar{X}_i$ is one. Here then $d = j-i$ and the translation is needed since the $\bar{X}_i$ and y-axes are $(i-1)$ units apart. Hence, ℓ is represented by $N-1$ such points where the indexing is essential specifying the linear relations between the variable pairs x_i and x_j and also needed as input to algorithms (e.g. finding the minimum distance between pairs of lines [3]). An important consequence of Desargues theorem is that for any $i \neq j \neq k$ the 3 points $\bar{\ell}_{ij}$, $\bar{\ell}_{jk}$, $\bar{\ell}_{ik}$ are collinear Fig. 2. In particular for a line $\ell \subset \mathbb{R}^3$ the $\bar{\ell}_{12}$, $\bar{\ell}_{13}$, $\bar{\ell}_{23}$ are always on a line $\bar{L}$. The 3-point-collinearity property has higher dimensional generalizations.

The $\bar{\ell}_{i,i+1}$ $(j = i+1)$ are the most commonly used representing points. A polygonal line $\bar{P}$ on the $N-1$ $\bar{\ell}_{i,i+1}$ necessarily represents a point $P = (p_1, ..., p_{i-1}, p_i, ..., p_N) \in \ell$ since the pair of values p_{i-1}, p_i simultaneously satisfy eq. (4) for every pair $i = 2, ..., N$. In Fig. 3 several polygonal lines are seen intersecting at the $\bar{\ell}_{i-1,i}$ representing a line $\ell \subset \mathbb{R}^{10}$. That is, both the line and its

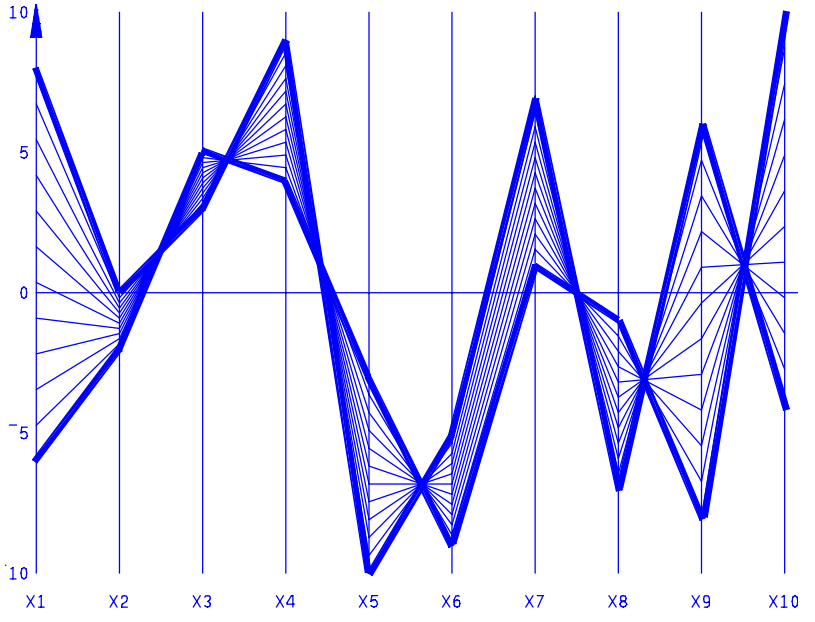


Fig. 3. Interval on a line $\ell \subset \mathbb{R}^{10}$, the polygonal lines intersect at 9 points $\bar{\ell}_{i,i+1}$, $i = 1, \ldots, 8$ which provide the *complete* description of ℓ. The point $\bar{\ell}_{12}$ being to the right of $\bar{X}_2$-axis $\Rightarrow$ slope of ℓ_{12} is $\in [0, 1]$, ℓ_{67} has slope 1 since the corresponding lines are parallel. The remaining $\ell_{i,i+1}$ have negative slopes with their $\bar{\ell}_{i,i+1}$ being in between the axes.

points are visualized. The representation of a line in terms of $N - 1$ points still holds when some x_i are constant [4] and occurs in the hypercube's representation Fig. 5.

2.3 Planes, Hyperplanes and Recursion

Vertical Line Representation. While a line can be determined from its projections, even in 3-D it is not possible to identify a (full) plane from two arbitrary projections. By contrast, coplanarity is nicely characterized with ∥-coords. A plane π shown in Fig. 4(left), intersects the x_1x_2 and x_2x_3 planes at the lines y^1 and y^2 respectively with $A = y^1 \cap y^2$. Let Y_i be the family of lines parallel to y^i , $i = 1, 2$. Each y^i being a line in $\mathbb{R}^3$ is represented by two points the $\bar{y}^i_{12}$ and $\bar{y}^i_{23}$. To simplify, here $\bar{y}^1_{12} = \bar{y}^1$, $\bar{y}^2_{23} = \bar{y}^2$ with the other two points on the x-axis not shown. A non-orthogonal coordinate system on π is formed, using the y^i as axes, so that a point P is determined as the intersection of two lines parallel to y^1 and to y^2 respectively. As pointed out in subsection 2.1, each of the families of lines Y_i is represented by a vertical line $\bar{Y}_i$ containing the point $\bar{y}_i$. Actually $\bar{Y}_i$ represent the *projections* of the parallel lines on the x_1x_2 and x_2x_3 planes respectively. By choosing a point $A \in \pi$ a distinct plane is specified

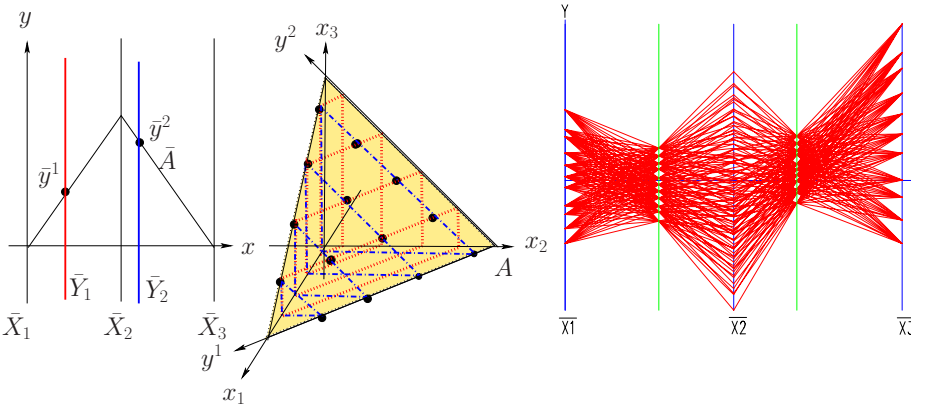


Fig. 4. (left) A plane π in $\mathbb{R}^3$ represented by two vertical lines and a polygonal line. A set of coplanar grid points – note the two vertical lines pattern (right).

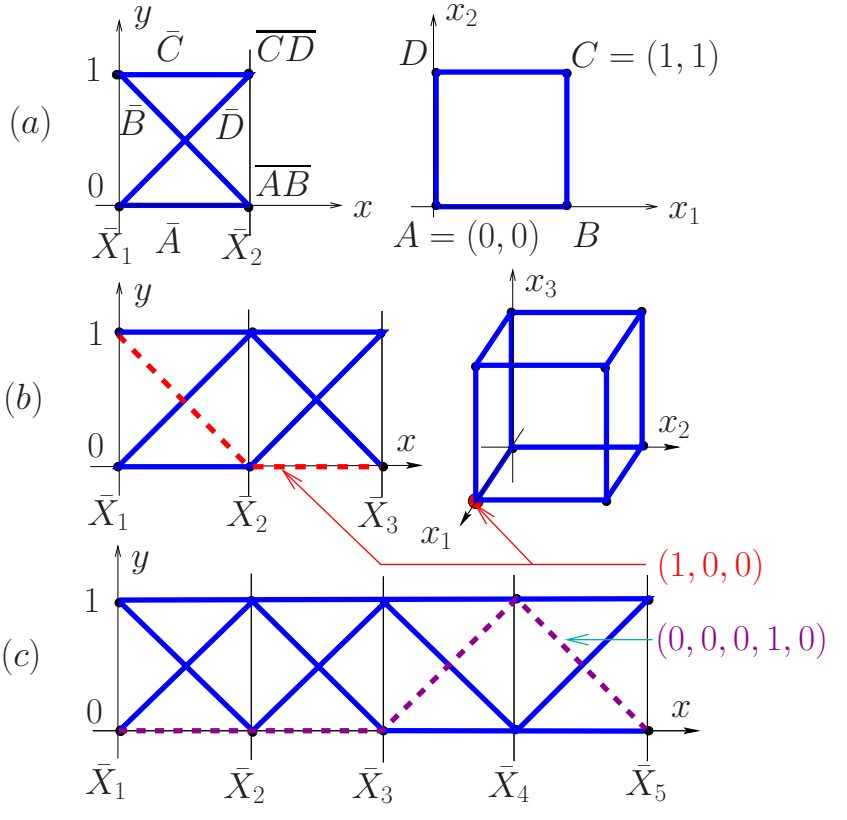


Fig. 5. (a) Square (b) Cube in $\mathbb{R}^3$ (c) Hypercube in $\mathbb{R}^5$ – all edges have unit length

having $\bar{Y}_1 \,.\, \bar{Y}_2$ as its $\parallel$ coordinate system and $\bar{A}$ its origin. Clearly the same argument can be extended to any dimension concluding that a hyperplane in $\mathbb{R}^N$, which is an $N-1$ dimensional flat, can be represented by a $N-1$ vertical lines and a polygonal line representing one of its points. Conversely, a set of coplanar points chosen on a grid, such as the one formed by lines parallel to a coordinate system, is represented by polygonal lines whose pattern specifies two vertical lines as shown in Fig. 4 (right).

The representation of a hypercube Fig. 5 is a preview of coming attractions. A square (a) is displayed in Cartesian and $\parallel$-coords dually mapping a vertex A into a line $\bar{A}$ and an edge AB into a point $\overline{AB}$. A cube (b) in 3-D, positioned in the first quadrant with a vertex at the origin and axes-parallel sides of unit length, is represented by *twice the square's pattern* the vertices' coordinates being all ordered triples of 0's and 1's. The edges are found by the intersection of the polygonal lines representing the two vertices. For example, the two points at $(1,0)\,,\,(2,0)$ represent the edge on the vertices $(0,0,0)\,,\,(1,0,0)$. The polygonal lines corresponding to these vertices together with the $\bar{X}_2\,,\,\bar{X}_3$ axes represent the faces (planes) at $x=0\,,\,x=1$. Here the two vertical lines representing the planes coincide with the parallel axes. All the cube's vertices, edges and faces are ascertained (i.e. are "visible") from the pattern where any interior polygonal line (i.e. within the rectangle $x \in [0,2]\,,\,y \in [0,1]$) represents one of the cube's interior points. The representation of a hypercube in $\mathbb{R}^5$, with one vertex at the origin and axes-parallel edges, is 4 times the square's pattern and all vertices, edges and faces of all order can be determined from the picture and best viewed interactively the same way as for the 3-D cube. The pattern's repetition pleasingly reveals the hypercube's symmetries and relation to the square. and the first an instance of a hypersurface's representation by linked planar regions. The ease of generalization from 3-D to higher dimensions is "built-in" in $\parallel$-coords and also applies to the representation of polytopes [5] and much more complex objects to be seen in the ensuing.

Recursion. While it is easy detecting coplanarity of points on a regular grid, for randomly chosen coplanar points no pattern is apparent Fig. 6 (left). A new approach is called for [6]. Rather than its points, an M dimensional hyperplane is represented in terms of its $M-1$ flats which are in turn represented by their $M-2$ flats and so on. Starting from its points, a recursive construction algorithm builds the hyperplane's flats increasing the dimension by one at each step. Let's see how it works for the points on the plane π in Fig. 6 (left). Lines on π are constructed from pairs of polygonal lines providing the line $\bar{L}$ of the 3 point collinearity described above and shown in Fig. 2. The result seen on Fig. 6 (right) is stunning. All the $\bar{L}$ lines intersect at a point which turns out to be characteristic of coplanarity but not enough to specify the plane π. Translating the first axis $\bar{X}_1$ to the position $\bar{X}_{1'}$, one unit to the right of the $\bar{X}_3$ axis and repeating the construction, for the $\bar{X}_2, \bar{X}_3, \bar{X}_{1'}$ coordinate system, yields a second point seen in Fig. 7 (left). These two points suffice to identify the plane π. They are denoted by $\bar{\pi}_{123} = \bar{\pi}_{0'}\,,\, \bar{\pi}_{1'23} = \bar{\pi}_{1'}$ having *3 indices*, since the plane's equation involves 3 variables, and are distinguished from the points representing

a line which have *2 indices*. Two more points are similarly obtained (translating the $\bar{X}_2$ and then $\bar{X}_3$ axes 3 units to the right). From the 4 points, Fig. 7 (right), the plane's equation can be read from the picture! Just the two points are needed to characterize the plane uncluttered by polygonal lines.

A few words are in order about the nice geometry behind this result. The family of planes, called *super-planes* (abbr. *sp*):

$$\pi^s \ : \ (x_3 - x_2) + k(x_2 - x_1) = 0 \ , \ k = \frac{d_2 - d_3}{d_2 - d_1} \ . \tag{6}$$

are on the line u through the points $(0,0,0)$ and $(1,1,1)$. The ratio k is determined by the axes spacing where the $\bar{X}_i$ are placed at the distance d_i from the y-axis. In this axes system points of the associated *sp*, determined by k, are represented by straight lines (try it). With the y and $\bar{X}_1$ axes coincident, $d_1 = 0$, $d_2 = 1$, $d_3 = 2$ specifying the *sp* (called the "first super-plane") given by $\pi_1^s : x_1 - 2x_2 + x_3 = 0$. As already mentioned, its points are represented by *straight* lines in the $\bar{X}_1, \bar{X}_2, \bar{X}_3$ coordinate system and all points so represented belong to π_1^s. Hence a line on π_1^s even though it is 3-dimensional is represented by a *single point*; the intersection of two straight lines (representing points of π_1^s). Further, the line $\bar{L}$ in the 3-pt-collinearity, Fig. 2, represents a point on the line $\ell \subset \pi$ being on the 3 points $\bar{\ell}_{12}, \bar{\ell}_{13}, \bar{\ell}_{23}$. Since it is represented by a *straight* rather than a polygonal line, $L = \pi \cap \pi_1^s$. Another line $\ell' \subset \pi$ constructed from a pair of polygonal lines determines another $L' \in pi \cap \pi_1^s$. The points L, L' are on the line $\ell_\pi = \pi \cap \pi_1^s$, seen in Fig. 8, represented by the single point in Fig. 6 (right).

The translation $\bar{X}_1 \rightarrow \bar{X}_{1'}$, $d_3 = 0 \rightarrow d_{3'} = 3$ specifies the second *sp* $\pi_{1'}^s \ : \ x_1 + x_2 - 2x_3 = 0$, corresponding to a 120^o rotation of $\pi_1^s \rightarrow \pi_{1'}^s$

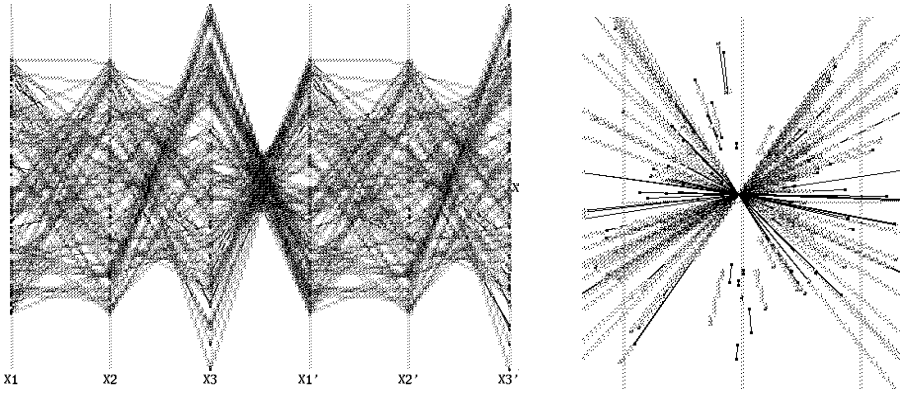


Fig. 6. (left) The polygonal lines on the first 3 axes represent a randomly chosen set of coplanar points. (right) Seeing coplanarity! Constructing lines on the plane, with the 3 point collinearity property, yields intersection at a point.

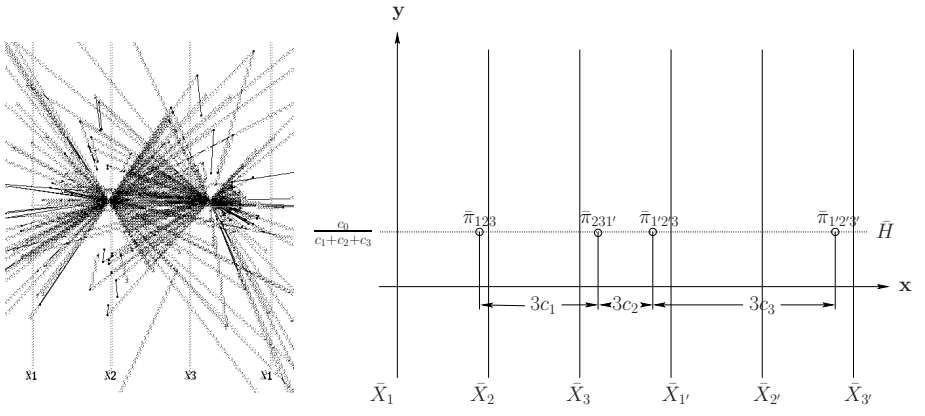


Fig. 7. (Left)The two points where the lines intersect uniquely determine a plane π in 3-D.(Right) From four points, constructed by consecutive axes translations, the normalized (i.e. $c_1 + c_2 + c_3 = 1$) coefficients of $\pi : c_1x_1 + c_2x_2 + c_3x_3 = c_0$ can be read from the picture!

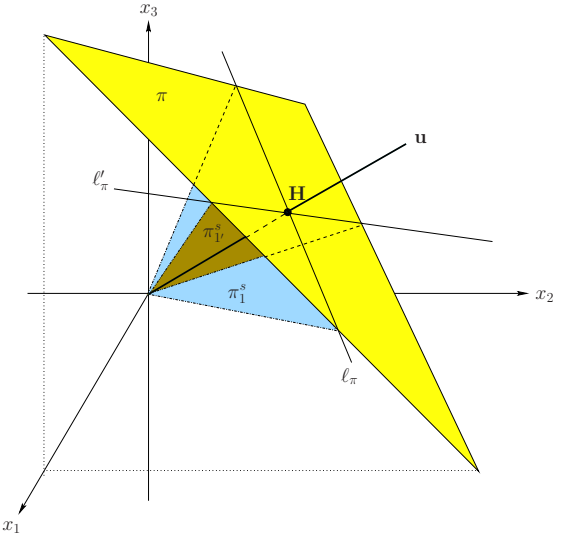


Fig. 8. The intersections of a plane π with the two super-planes ${\pi^s}_1$ and $\pi^s_{1'}$. These are the two lines ℓ_π , ℓ'_π which specify the plane, represented by two points as in Fig. 7 (right), and provide its representation.

about u. Repeating the construction for the $\bar{X}_2, \bar{X}_3, \bar{X}_{1'}$ axes system produces the second point in Fig. 7 (left) and equivalently the second intersecting line $\ell'_\pi = \pi \cap \pi^s_{1'}$ seen in Fig. 8. The lines ℓ_π, ℓ'_π intersect on u and specify the plane π. In retrospect for the constructions in Fig. 6 and Fig. 7, just two intersecting

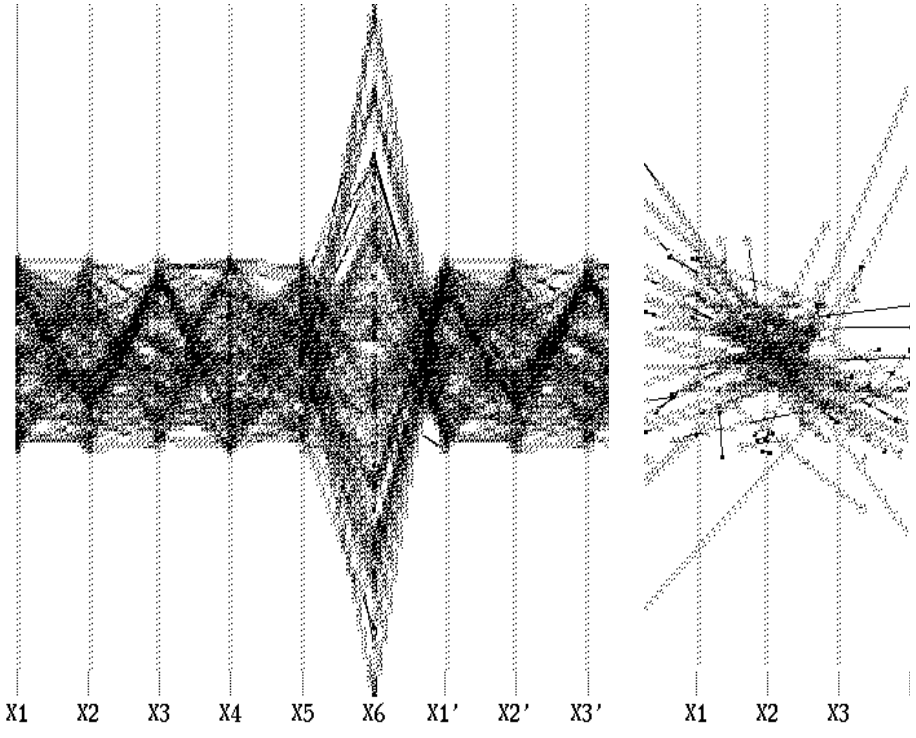


Fig. 9. Randomly selected points on a hyperplane π^5 (a 5-flat) in $\mathbb{R}^6$. The $\bar{\pi}_{12}$, $\bar{\pi}_{23}$ portions of the 1-flats $\subset \pi^5$ constructed (right) from the polygonal lines. No pattern is evident. nor for the subsequent construction (not shown) of the points $\bar{\pi}^k$ representing 2 and 3-flats of π^5.

lines are needed to determine the points. The x coordinates of the 3rd and 4th points in Fig. 7 are obtained from $x_{2'} = 6 - (x_{0'} + x_{1'})$, $x_{3'} = 3 + x_{0'}$ where $x_{i'}$ denotes the x coordinate of $\bar{\pi}_{i'}$.

The gist is that while coplanar points can not be visually detected, a plane can be recognized from its **lines** in $\mathbb{R}^3$ and a hyperplane from its $N-2$-flats in $\mathbb{R}^N$. In general, a hyperlane in N-dimensions is represented uniquely by $N-1$ points each with N indices. The algorithm constructs these points *recursively*, raising the dimensionality by one at each step, as is done here starting from points (0-dimensional) constructing lines (1-dimensional) and so on. An example is shown in Figs. 9 and 10 constructing the representing points for a hyperplane in $\mathbb{R}^6$.

All the nice higher dimensional projective dualities like *point* $\leftrightarrow$ *hyperplane*, *rotation* $\leftrightarrow$ *translation* etc hold. Further, a multidimensional object, represented in ‖-coords, can still be recognized after it has been acted on by projective transformations (i.e. translation, rotation, scaling and perspective). The recursive construction and its properties are at the heart of the ‖-coords methodology.

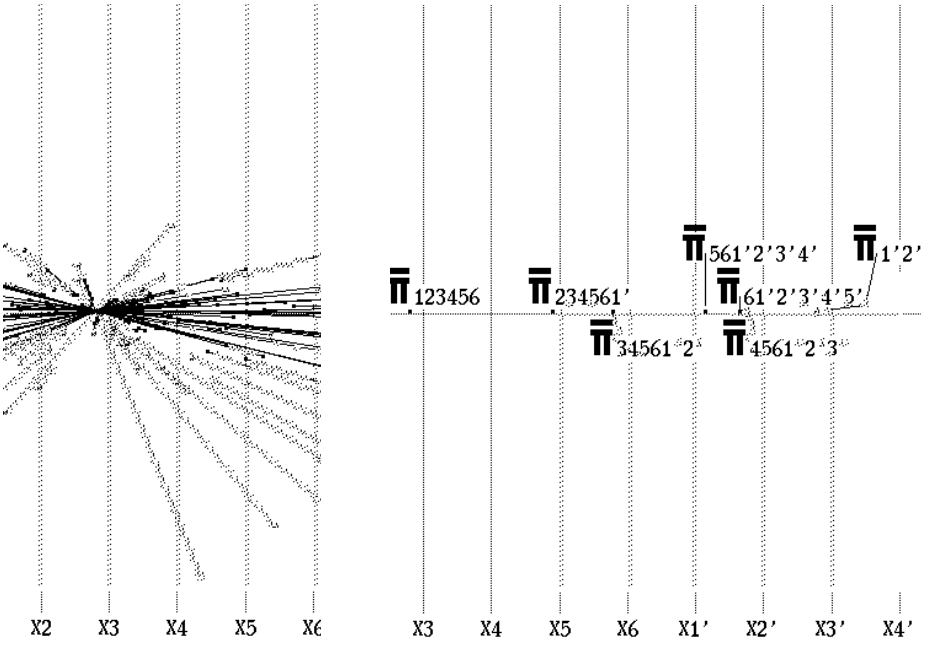


Fig. 10. This is it! On the left are the $\bar{\pi}^4_{12345}$, $\bar{\pi}^4_{23456}$ of a 4-flat $\subset \pi^5$ constructed from the polygonal lines joining $\bar{\pi}^3_{1234}$, $\bar{\pi}^3_{2345}$, $\bar{\pi}^3_{3456}$ of two 3-flats in π^5. This shows that the original points in Fig. 9(left) are on a 5-flat in $\mathbb{R}^6$. The remaining points of the representation are obtained in the same way and all 5 representing points of π^5 plus the two additional ones as in Fig. 7 (right) are seen on the right. The coefficients of π^5's equation are equal to 6 times the distance between sequentially indexed points.

Translation ↔ Rotation. The *rotation* ↔ *translation* correspondence between ∥ and Cartesian coordinates works for flats in $\mathbb{R}^N$. It is illustrated in Fig. 11 for $\mathbb{R}^3$ with the rotation of a plane π^2 about a line π^1. The superscripts indicate the flats' dimensionality. When it is clear from the context, the superscripts are dropped for the dimensionality is apparent from the number of indices. For any plane π the four points $\bar{\pi}_{123}$, $\bar{\pi}_{231'}$; , $\bar{\pi}_{31'2'}$, $\bar{\pi}_{1'2'3'}$ lie on a horizontal line $\bar{H}$. For the line π^1, as previously mentioned, the 3 points $\bar{\pi}_{12}$, $\bar{\pi}_{13}$, $\bar{\pi}_{23}$ are on the same line $\bar{L}$. It is useful to include the representation of the line π^1 in terms of the other triples of axes. Namely, the points

$\bar{\pi}_{1'2}\, \bar{\pi}_{1'3}\, \bar{\pi}_{2'3}$ *on line* $\bar{L}'$, $\bar{\pi}_{12'}\, \bar{\pi}_{13'}\, \bar{\pi}_{2'3'}$ *on line* $\bar{L}''$, $\bar{\pi}_{12'}\, \bar{\pi}_{13'}\, \bar{\pi}_{2'3'}$ *on line* $\bar{L}'''$.

The point $\bar{\pi}_{123}$ is on the intersection of $\bar{H}$ with the line $\bar{L}$ and similarly

$$\bar{\pi}_{231'} = \bar{H} \cap \bar{L}' \,, \bar{\pi}_{31'2'} = \bar{H} \cap \bar{L}'' \,, \bar{\pi}_{1'2'3'} = \bar{H} \cap \bar{L}''' \,. \tag{7}$$

A rotation of the plane π^2 about the line π^1 becomes a vertical translation of the line $\bar{H}$ with $\bar{\pi}_{123}$ translated along the line $\bar{L}$; similarly the differently indexed

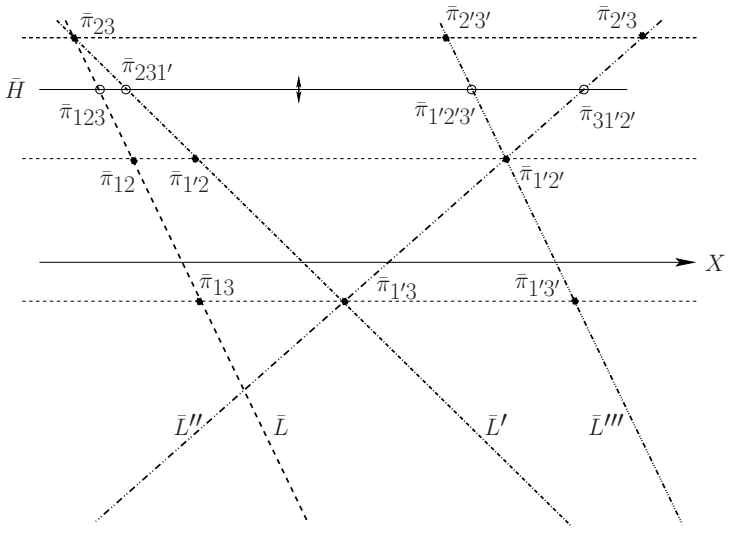


Fig. 11. Rotation of a 2-flat (plane) about a 1-flat(line) in $\mathbb{R}^3$ corresponds to a translation of its representing points (with 3 indices) on the horizontal line $\bar{H}$ along the lines $\bar{L}$, $\bar{L}'$, $\bar{L}''$, $\bar{L}'''$ joining the representing points (with 2 indices) representing the 2-flat.

points are translated along their corresponding lines $\bar{L}'$, $\bar{L}''$ and $\bar{L}'''$ as dictated by eq. (7). At any instance, the (directed) distance between adjacent points is equal to the coefficients of the plane's equation as pointed out in Fig. 7 (right). Where pairs of $\bar{L}$-lines intersect one coefficient is zero and π^2 is perpendicular to a principal 2-plane. Specifically, at

$$\begin{aligned}
&\bar{H} \cap \bar{L} \cap \bar{L}' , \quad c_1 = 0 \ \textit{and} \ \pi^2 \perp x_2x_3 - \textit{plane} , \\
&\bar{H} \cap \bar{L}' \cap \bar{L}'' , \ c_2 = 0 \ \textit{and} \ \pi^2 \perp x_1x_3 - \textit{plane} , \\
&\bar{H} \cap \bar{L}'' \cap \bar{L}''' , \ c_3 = 0 \ \textit{and} \ \pi^2 \perp x_1x_2 - \textit{plane} .
\end{aligned}$$

Consider π as an oriented plane with normal vector $\boldsymbol{N}$, whose components are the coefficients c_i. As vertically translating $\bar{H}$ crosses a pairwise intersection of $\bar{L}$ a c_i changes sign. So after traversing *all 3* $\bar{L}$ intersections the oriented plane completes a 180^0 rotation i.e. the normal $\boldsymbol{N} \rightarrow -\boldsymbol{N}$. This is an informal description of the "flipping lemma" revisited later in subsection 5 in connection with the non-orientability of the Möbius strip.

2.4 Curves

The image $\bar{c}$ of a point-curve

$$c \, : \, F(x_1, x_2) = 0 \ \mapsto \ \bar{c} : f(x, y) = 0 \tag{8}$$

can be obtained as a *line-curve* the lines images of points of c. The better way is to map the tangents of c directly into points obtaining a *point-curve* image $\bar{c}$ whose point coordinates are given by:

$$\begin{cases} x = d\frac{\partial F/\partial x_2}{(\partial F/\partial x_1 + \partial F/\partial x_2)} & , \\ y = \frac{(x_1 \partial F/\partial x_1 + x_2 \partial F/\partial x_2)}{(\partial F/\partial x_1 + \partial F/\partial x_2)} & . \end{cases} \tag{9}$$

obtained directly from eq. (2) with $m = -\frac{\partial F/\partial x_1}{\partial F/\partial x_2}$ with inter-axes distance d. The zeros of the denominator provide the ideal points, if any, of $\bar{c}$ where the tangent has slope 1. For conics F is quadratic and the the partial derivatives in eq. (9) are linear. So $(x_1, x_2) \mapsto (x, y)$ is a Möbius transformaton and so is its inverse [7]. Therefore, the inverse relation of (x_1, x_2) in terms of (x, y) is also quadratic showing that

$$Conics \leftrightarrow Conics \; . \tag{10}$$

generalizing to higher-dimensional quadric hypersurfaces. Curves play a crucial role in the ensuing being the boundaries of the linked planar regions representing surfaces. For algebraic curves, as in Fig. 12, Plücker's formulae [8] are very useful. A couple of examples, Figs. 12 and 13, are in order.The dualities *cusp* $\leftrightarrow$ *inflection-point*, *crossing-point* $\leftrightarrow$ *bitangent* hold and in fact carry over to hypersurfaces.

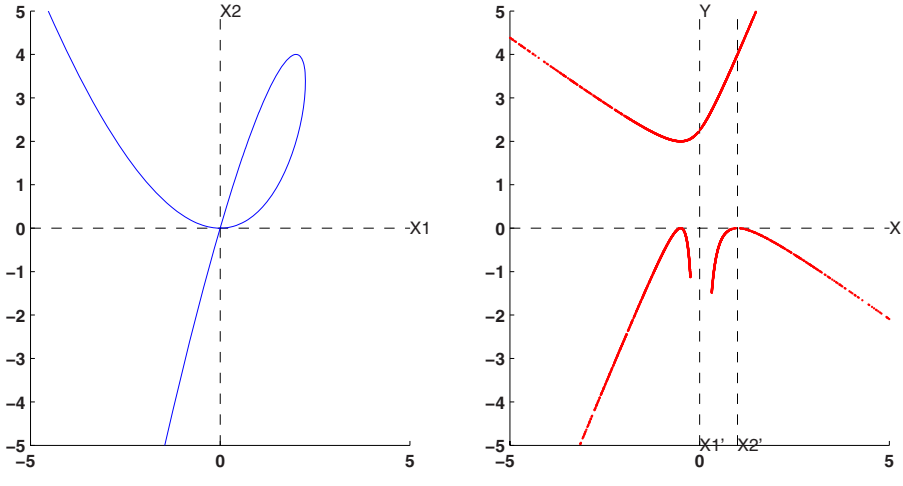


Fig. 12. The image of the curve $c : F(x_1, x_2) = x_1^3 + x_1^2 - 3x_1x_2 = 0$ has degree 4 (determined from Plücker's formula). This is an example of the *crossing-point* $\leftrightarrow$ *bitangent* duality.

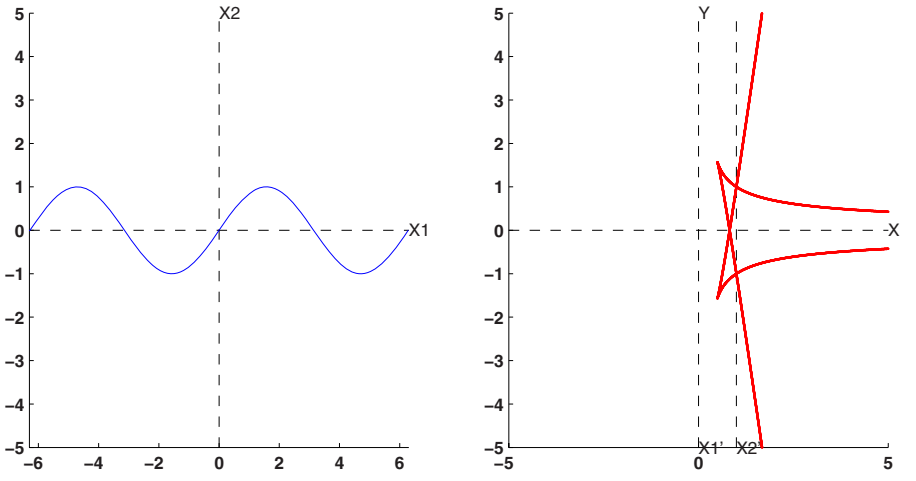


Fig. 13. The image of $c : x_2 = sin(x_1)$ for $x_1 \in [-2\pi, 2\pi]$. A periodic curve has image symmetric about the x-axes. This example illustrates the *ip – inflection point* $\leftrightarrow$ *cusp* duality. The *ip* at $x_1 = \pm\pi$ are mapped into the two cusps, and the *ip* at the origin having slope $m = 1$ is mapped to the ideal point along the x-axis.

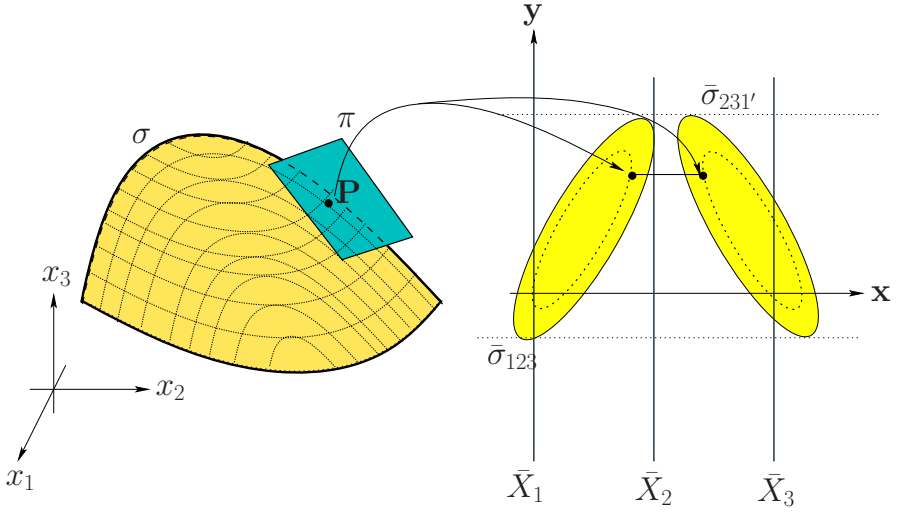


Fig. 14. A surface $\sigma \in \mathcal{E}$ is represented by two planar regions $\bar{\sigma}_{123}$, $\bar{\sigma}_{231'}$. consists of the pairs of points representing all its tangent planes.

3 Surface Representation – Formulation

The representation is developed for the family $\mathcal{E}$ of smooth hypersurfaces in $\mathbb{R}^N$ which are the envelopes of their tangent hyperplanes When the

higher-dimensional generalizations are clear the discussion is confined to $\mathbb{R}^3$. Fig. 14. At each point $P \in \sigma$, the tangent plane π is mapped to the two planar points $\bar{\pi}_{123}$, $\bar{\pi}_{231'}$. Collecting the points by their indices for all the tangent planes yields two planar regions $\bar{\sigma}_{123}(x, y)$, $\bar{\sigma}_{1'23}(x, y)$ one each for the index triples. The regions need to be *linked* via a *matching algorithm* which selects the pairs of points (one from each region) representing valid tangent planes of σ. The two linked regions form the representation $\bar{\sigma}$ of the surface. This is the extension of the linking already due to the indexing of the representing points for lines and hyperplanes. The intent is to *reconstruct* the surface from its representation. The manner and extent to which this is possible, i.e. when the ‖-coords mapping is invertible, are the issues studied. Formally, for a surface $\sigma \in \mathcal{E}$ given by

$$\sigma \ : \ F(\mathbf{x}) = 0 \ , \ \mathbf{x} = (x_1, x_2, x_3) \tag{11}$$

its representation in ‖-coords is

$$\sigma \mapsto \bar{\sigma} = (\bar{\sigma}_{123}(x, y) \ , \ \bar{\sigma}_{1'23}(x, y)) \subset \mathbb{P}^2 \times \mathbb{P}^2 \tag{12}$$

where use of the projective plane $\mathbb{P}^2$ allows for the presence of ideal points. In the notation the *link* between the two regions is indicated by placing them within the (,) which are omitted when the discussion pertains to just the region(s). The functions used are assumed to be continuous in all their variables together with such of its derivatives as are involved in the discussion.

The gradient vector of F, $\nabla F = (\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \frac{\partial F}{\partial x_3})\big|_P$, at the point P is normal to the surface σ at P and so the the tangent plane π of σ at the point $P_0(\mathbf{x^0})$, $(\mathbf{x^0}) = (x_1^0 \ , x_2^0 \ , x_3^0)$, is given by

$$\pi : \nabla F \cdot (\mathbf{x} - \mathbf{x^0}) = \sum_{i=1}^{3} (x_i - x_i^0) \frac{\partial F}{\partial x_i}(x_1^0, x_2^0, x_3^0) = 0 \ . \tag{13}$$

At times it is preferable to describe a surface in terms of 2 parameters as

$$\sigma \ : \ F(s, t) = F(\mathbf{x}) = 0 \ , \quad \mathbf{x} = \mathbf{x}(\mathbf{s}, \mathbf{t}) \ , \ s \in I_s \ , \ t \in I_t \ , \tag{14}$$

where I_s , I_t are intervals of $\mathbb{R}$ with the representing points of π

$$\bar{\pi}_{i'}(s, t) = (\nabla F \cdot \mathbf{d^i} \ , \ \nabla F \cdot (\mathbf{x^0}) \ , \ \nabla F \cdot \mathbf{u}) \ , \quad i = 0, 1. \tag{15}$$

The equivalent description for hypersurfaces in $\mathbb{R}^N$ requires $N - 1$ parameters. This parametric form is due to Gauss. The vector description is in many ways simpler, clarifying the surface's structure and higher-dimensional generalization, is due to the mathematical physicist W.J. Gibbs who was an early advocate for *visualization* in science[2].

[2] In 1873 he wrote the two papers: "Graphical Methods in the Thermodynamics of Fluids" and "A Method of Geometrical Representation of the Thermodynamics Properties of Substances by Means of Surfaces".

Stated explicitly the point $\mapsto$ pair-of-points mapping,

$$\mathbf{x} \in \sigma \mapsto \pi \mapsto (\bar{\pi}_{0'}, \bar{\pi}_{1'}) = (\bar{\pi}_{123}\ ,\ \bar{\pi}_{1'23}) = ((x_{0'}, y), (x_{1'}, y))\ , \tag{16}$$

the $x = x_{0'}$ and $x' = x_{1'}$ are the x-coordinates of $\bar{\pi}_{1'23}$, $\bar{\pi}_{1'23}$ respectively. The y is the same for both points. In terms of the gradient's components, $F_i = \partial F/\partial x_i$

$$\begin{cases} x_{0'} = \frac{F_2+2F_3}{F_1+F_2+F_3}\ , \\ y = \frac{x_1F_1+x_2F_2+x_3F_3}{F_1+F_2+F_3}\ , \\ x_{1'} = \frac{3F_1+F_2+2F_3}{F_1+F_2+F_3}\ . \end{cases} \tag{17}$$

These transformations are the direct extension of the 2-D point $\leftrightarrow$ point curve transformations eq. (9) A word of caution, when the inter-axes distance $d \neq 1$, the right-hand-sides of $x_{0'}$ and $x_{1'}$ above need to be multiplied by d and $2d$ respectively (see eq. (9)).

3.1 Boundary Contours

The construction of the regions $\bar{\sigma}$, and in particular their boundary $\partial\bar{\sigma}$, is greatly facilitated by the next lemma. To guarantee that the intersections below are non-empty the origin is, translated if necessary, to an interior point of the surface σ.

Lemma 1. *(Boundary of $\bar{\sigma}$) For σ a surface in $\mathbb{R}^3$, $\partial\bar{\sigma}_{123}$ is the image of $\sigma \cap \pi^s_{0'}$. That is $\partial\bar{\sigma}_{123} = \overline{\sigma \cap \pi^s_{0'}}$ and similarly $\partial\bar{\sigma}_{1'23} = \overline{\sigma \cap \pi^s_{1'}}$.*

Proof. With reference to Fig. 15 each point $\bar{\rho}_{123} \in \bar{\sigma}_{123}$ is a representing point of a plane ρ tangent to the surface σ at a point R. In fact, $\bar{\rho}_{123} = \bar{\ell}_\rho$ where $\rho \cap \pi^s_{0'} = \ell_\rho$. In addition, for a plane π tangent to σ at a point P where $\bar{\pi}_{123} \in \partial\bar{\sigma}_{123}$ then the corresponding $\ell_\pi = \pi \cap \pi^s_{0'}$, $P \in \ell_\pi$. The image $\bar{P}$ is a line tangent to $\partial\bar{\sigma}_{123}$ at $\bar{\pi}_{123} = \bar{\ell}_{123} \Rightarrow P \in \pi^s_{0'}$. In short, every point of $\bar{\sigma}_{123}$ represents a line of $\pi^s_{0'}$ due to the smoothness of $\partial\sigma$ and, therefore, continuity of $\partial\bar{\sigma}_{123}$, every tangent to $\partial\bar{\sigma}_{123}$ represents a point of $\pi^s_{0'}$ and $\partial\sigma$. Hence $\partial\bar{\sigma}_{123} = \overline{\sigma \cap \pi^s_{0'}}$ and similarly $\partial\bar{\sigma}_{1'23} = \overline{\sigma \cap \pi^s_{1'}}$. The smoothness of $\partial\sigma$ ensures the continuity of $\partial\bar{\sigma}_{123}$.

The boundary $\partial\bar{\sigma}$ is simply the image of two *curves*: the intersections of σ with the first two superplanes $\pi^s_{0'}, \pi^s_{1'}$. Note further that for a plane π tangent at a point $P \in \sigma$, $\bar{\pi}_{123} \in \partial\bar{\sigma}_{123}$ and $\bar{\pi}_{1'23} \in \partial\bar{\sigma}_{1'23} \Leftrightarrow P \in \sigma \cap \pi^s_{0'} \cap \pi^s_{1'}$.

Lemma 2. *(Boundary of $\bar{\sigma}$ in $\mathbb{R}^N$) For a $\sigma \subset \mathbb{R}^N$, $\partial\bar{\sigma}$ is composed of $N-1$ curves which are the images of the intersections of σ with the first $N-1$ superplanes.*

An algebraic surface is one described by a polynomial equation providing an important special case of Lemma 2.

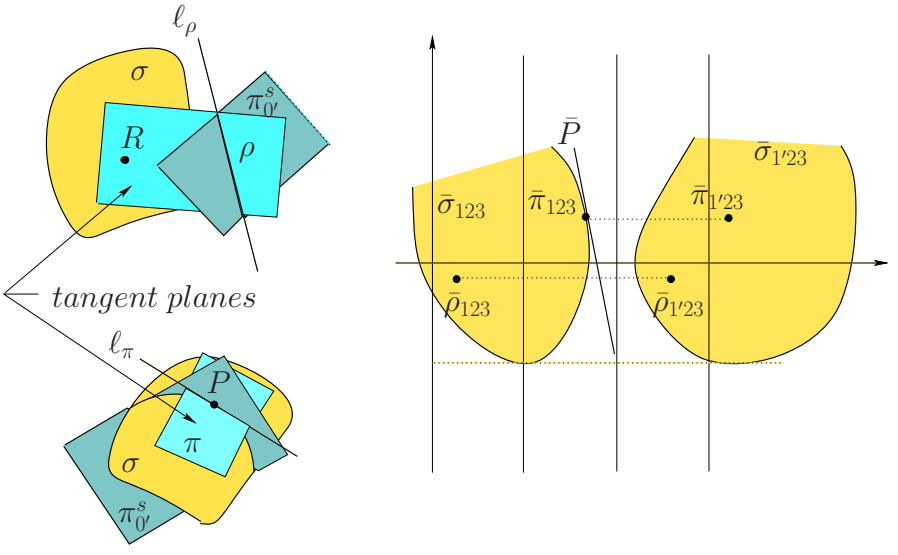


Fig. 15. Formation of boundary contour

Corollary 1. *(Boundary of an algebraic surface) The boundary $\bar{\sigma}$ of an algebraic surface σ of degree n is composed of $N-1$ algebraic curves each of degree $n(n-1)$ or less.*

The corresponding known result in Algebraic Geometry is that the dual of a non-singular algebraic surface of degree n has degree $n(n-1)^{(N-1)}$ [9], [10]. Here the boundary representations can be found with the aid of *Plücker's* formulae. From this lemma we can conclude that the for quadric surfaces (i.e. those whose degree is 2) their boundary contours are conics.

Corollary 2. *(Ideal points) An ideal point on the boundary $\partial\bar{\sigma}$ is the image of a tangent plane, at a point of σ, parallel to* **u**.

Proof. It is an immediate consequence of eq. (15) that ideal points occur in the image when the 3rd component is zero. That is the tangent plane's normal is normal to **u** or the tangent plane is parallel to **u**.

The ideal points map into asymptotes determining the branches of the boundary curves. The boundary $\partial\bar{\sigma}$ is the image of *space* curves contained in the super-planes. A space curve can be represented as the image of its tangent lines. In turn, the tangent t at point $P \in c$ can be represented by $N-1$ doubly indexed points say $\bar{t}_{i-1,i}$ so that $\bar{c}$ consists of $N-1$ planar curves $\bar{c}_{i-1,i}$ formed by the $\bar{t}_{i-1,i}$; two such curves for $\mathbb{R}^3$. Each $\bar{c}_{i-1,i}$ is the image of the projection of c on the 2-plane $x_{i-1,i}$ The fun part is that, in our case, these two curves are coincident since they are the images of a curve in a super-plane say $c \subset \pi^s_{0'}$ where each tangent t at a point $P \in c$ is a line in π^s_1, and hence $\bar{t}_{12} = \bar{t}_{23}$.

4 Developable Surfaces

A smooth surface σ in R^3 generally has a 2-parameter family of tangent planes so that its ||-coord representation consists of two indexed planar regions $\bar{\sigma}_{123}$, $\bar{\sigma}_{1'23}$. Of particular interest is the special case when σ has a 1-parameter family of tangent planes. Then the $\bar{\sigma}_{123}$, $\bar{\sigma}_{1'23}$ degenerate into two curves. Surfaces with this property are precisely the class of *developables*, that is they can be unfolded(developed) into a plane without streching or contracting. Examples are cylinders, cones and their generalizations as well as more complex surfaces like the helicoid in Fig. 20.

As we shall see the simplicity of developable surfaces results in many pleasant properties of their ||-coord representations. A simple procedure exists to reconstruct the surface given an indexed pair of monotonic pieces of its representation. These pieces can then be strung together with some additional auxiliary pairs of marks. The representation is indeed unique for a large subclass of developable surfaces covering complex enough surfaces even if we restrict our attention to cases where the number of marks is finite.

4.1 Reconstruction

The reconstruction procedure can be summarised in a few sentences. Pick a point on one representing curve and find the other corresponding point (point at the same height) on the other representing curve. Then two tangent lines at these point determine a ruling of the surface enabling its reconstruction.

Consider a cone σ and two of its tangent planes $\pi(t_0)$ and $\pi(t)$ as shown on the left of Fig. 16 where the family of tangent planes of σ is parameterized by t. On the right is its ||-coord representation, a pair of curves $\bar{\sigma}_{123}$ and $\bar{\sigma}_{231'}$ instead of a pair of regions since σ is a developable surface. The secant line $l(t, t_0)$

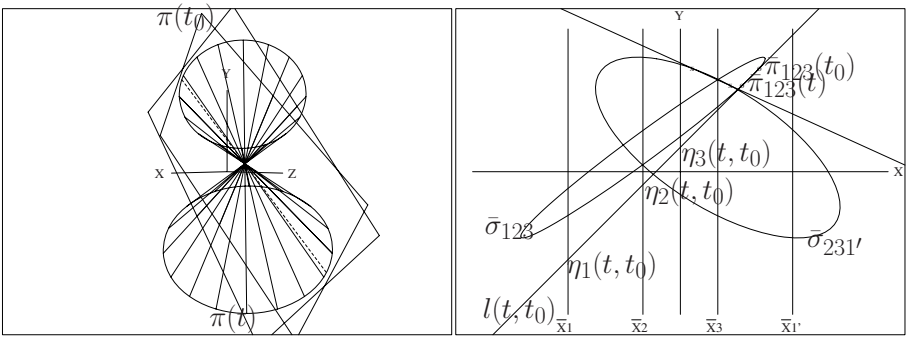


Fig. 16. The intersection of the two tangent planes $\pi(t)$ and $\pi(t_0)$ approaches the ruling of tangency $\rho(t_0)$ at $\pi(t_0)$ as the tangent plane $\pi(t)$ approaches $\pi(t_0)$. Accordingly, the secant line $l(t, t_0)$ joining the representation of these two tangent planes approaches the tangent line of $\bar{\sigma}_{123}$ at the point $\bar{\pi}(t_0)$.

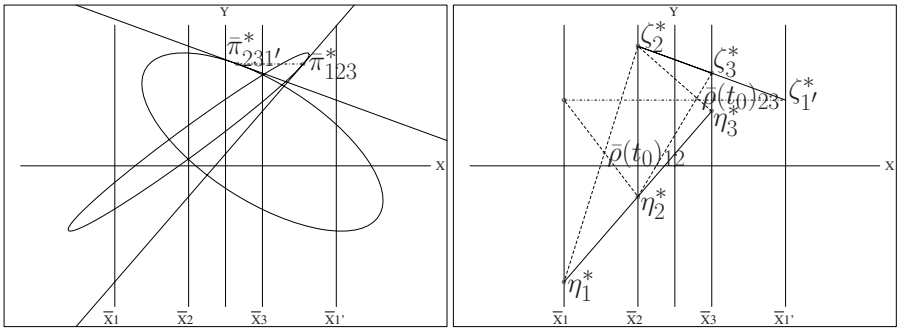


Fig. 17. The reconstruction procedure: Draw the tangent lines at the two corresponding points representing a tangent plane $\pi(t_0)$ to σ (left), find the intercepts of each line with the correct set of axes (either 123 or 231'), and finally join the corresponding pair of intercepts to obtain the two end points of the polygonal line representing a ruling $\rho(t_0)$. (right).

joining $\bar{\pi}_{123}(t_0)$ and $\bar{\pi}_{123}(t)$ necessarily represents the intersection (a straight line) of the two tangent planes $\pi_{123}(t_0)$ and $\pi_{123}(t)$ because of duality. Recalling the point representation of lines on the superplane π^s_1 the coordinates of the point in space $\eta(t, t_0) \equiv \pi(t) \cap \pi(t_0) \cap \pi^s_1$ are found by reading the intercepts of this secant line with the $\bar{X}_1$, $\bar{X}_2$, and $\bar{X}_3$ axes. As t approaches t_0, the line of intersection approaches $\rho(t_0)$ the line of tangency of $\pi(t_0)$ to σ and also a ruling of σ. In the $\|$-coord plane this corresponds to $\bar{\pi}_{123}(t)$ approaching $\bar{\pi}_{123}(t_0)$ and the secant line $l(t, t_0)$ approaching the tangent line of $\bar{\sigma}_{123}$ at $\bar{\pi}_{123}(t_0)$. Let's write $\eta^* \equiv \lim_{t \to t_0} \eta(t, t_0) = \rho(t_0) \cap \pi^s_1$, the intersection of the ruling of tangency at $\pi(t_0)$ with the special plane π^s_1, and write its coordinates $(\eta^*_1, \eta^*_2, \eta^*_3)$ as we read them off the intercepts.

Up to now we have focused on only one of the two representing curves of σ and obtained η^*. Repeating the same arguments on the other representing curve $\bar{\sigma}_{231'}$, we similarly obtain another point $\zeta^* \equiv \lim_{t \to t_0} \zeta(t, t_0) = \rho(t_0) \cap \pi^s_{1'}$, the intersection of the ruling of tangency with the other special plane. Fig. 17 shows the $\|$-coord representation of η^* as the three intercepts $(\eta^*_1, \eta^*_2, \eta^*_3)$, and that of ζ^* as the three intercepts $(\zeta^*_1, \zeta^*_2, \zeta^*_3)$. The ruling is now readily obtained by joining η^* and ζ^* in space, which translates to the multi-dimensional line construction in the $\|$-coord plane as illustrated in Fig. 3. A note of caution: the ζ^*_1 intercept was obtained on the $\bar{X}_{1'}$ but in order to construct the ruling, it has to be translated to the $\bar{X}_1$ axis before joining it with η^*_2.

We have just seen how to recover the developable surface from its $\|$-coord representation. This procedure is clearly general for it does not depend on any property of the cones. In fact we have simultaneously proved the uniqueness of the $\|$-coord representation for a developable surface provided its representing curves are monotonic with respect to the y axis.

4.2 Resolution of Ambiguity

It turns out that with the help of a few additional marks in the picture, the uniqueness result still holds valid given much weaker requirement on the representing curves than the obvious monotonic constraint as mentioned in the previous section.

For a cone, a cylinder, or any developable surface σ whose representation curves $\bar{\sigma}_{123}$ and $\bar{\sigma}_{231'}$ do not have any self-intersection points, the reconstruction procedure can be applied to a given pair of starting points $\bar{\pi}(t)_{123}$ $\bar{\pi}(t)_{231'}$ at $t = t_0$ and repeated as this pair of points are traced *smoothly* along the two curves. This means that, for example, when $\bar{\pi}(t)_{123}$ reaches an extremal point and enters the other Y-monotonic branch on $\bar{\sigma}_{123}$, we choose the corresponding $\bar{\pi}(t)_{231'}$ that also goes into the other Y-monotonic branch on $\bar{\sigma}_{231'}$. Thus there is no ambiguity in this case even if the representing curves are not Y-monotonic. Even for developable surfaces each of whose $\bar{\sigma}_{123}$ and $\bar{\sigma}_{231'}$ has some self-intersection points, there is no ambiguity as to which branch to follow at the intersections, as long as the branches don't share the same tangent line except in pairs, since we insist on *smooth* tracing. *Therefore the* ∥*-coord representation of developable surfaces, barring those whose representations have self-tangential points, is unique given just one pair of starting points in addition to* $\bar{\sigma}_{123}$ *and* $\bar{\sigma}_{231'}$.

For a σ whose representation has self-tangential points, we break $\bar{\sigma}_{123}$ and $\bar{\sigma}_{231'}$ at these self-tangential points into relatively simple pieces, each pair of which can be uniquely traced to reconstruct the entire surface, provided the pairing is known, say given by pairs of corresponding points each for one pair of pieces. In a way we are *patching* the developable surfaces from its pieces by such pairs of corresponding points (which of course cannot be self-tangential points themselves). If the number of pieces is finite, say k, then we don't even need the pairing information to be given *a priori*. A $O(k^2)$ pairing algorithm can be applied to find all possible surfaces as a result of different pairings arising from ambiguity. As is easily verified, the worst case scenario is achievable but is very unlikely since it implies that a non-negligible proportion of these pieces share not only the same number of extremal points but also the same extremal y-coordinates occurring in exactly the same sequence. As a side note we mention that self-tangency includes the case where a curve has two or more pairs of branches sharing the same asymptote as we have to take points at infinity into account.

We can conclude that the ∥-coord representation of developable surfaces is unique and provides sufficient information for recovering the surfaces for almost all practical purposes. The reconstruction is still possible even without a priori pairing information, though at a quadratically higher cost, for all but the surfaces whose representing curves have an infinite number of self-tangencies that break the curves into similar pieces, an infinite number of which have the same number of extremal points and the same extremal y-coordinates occurring in exactly the same sequence. It also provides valuable pointers for treating more complex surface classes.

4.3 Higher Dimensions

A developable hypersurface σ in higher dimensions is the envelope of a one-parameter family of hyperplanes. Each of these hyperplanes is one of its tangent hyperplane. All the results in this section are generalizable to developable hypersurfaces in dimensions $N > 3$. These generalized results are stated without proof below for readers who would like to turn these theories into practical implementations.

Denote the $N-1$ representing curves as $\bar{\sigma}_{\beth}$ where $\beth \in I$ is one of the indexing sets $\{1, 2, \ldots, N\}, \{2, 3, \ldots, N, 1'\}, \ldots, \{N-1, N, 1', 2', \ldots, (N-2)'\}$. If the $\bar{\sigma}_{\beth}$'s are monotonic with respect to the y-axis, then σ can be reconstructed as follows:

1. Draw a horizontal line to intersect the $\bar{\sigma}_{\beth}$'s at $N-1$ points $\bar{\pi}(t_0)_{\beth}$ which together specify a tangent hyperplane $\pi(t_0)$.
2. Draw $N-1$ tangent lines $l_{\beth}$ at each $\bar{\pi}(t_0)_{\beth}$ to each $\bar{\sigma}_{\beth}$. Each of these lines specifies the intersecting point of the "ruling" (an $(N-2)$-flat) of σ at $\pi(t_0)$ with the superplane $\pi^s_{\beth}$ of the corresponding index.
3. Find the N intercepts of each of the $N-1$ $l_{\beth}$ with the corresponding axes to obtain the coordinates of these points of intersection.
4. Draw horizontal lines in order to bring all numbers to the first set of axes $\{\bar{X}_1, \ldots, \bar{X}_N\}$.
5. Follow the recursive construction procedure in Sec. 2.3 to obtain the $(N-2)$-dimensional ruling.

In general when the representing curves are not necessarily monotonic with respect to the y-axis, we break the curves into k traceable pieces at self-tangential

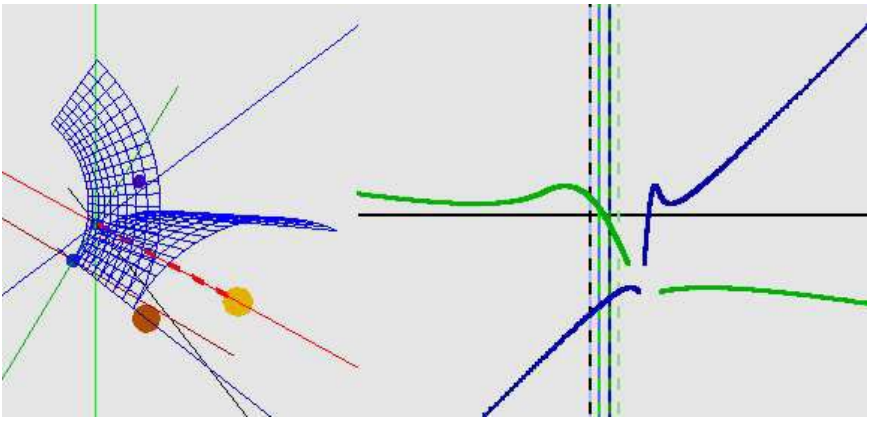


Fig. 18. A general cylinder illustrating the *developable* $\leftrightarrow$ *curve* duality. Here the two *inflection points* represent the ruling formed by *cusps* in $\mathbb{R}^3$. The equivalent hypersurface in $\mathbb{R}^N$ is represented by $N-1$ such curves all with inflection points at the same value of y.

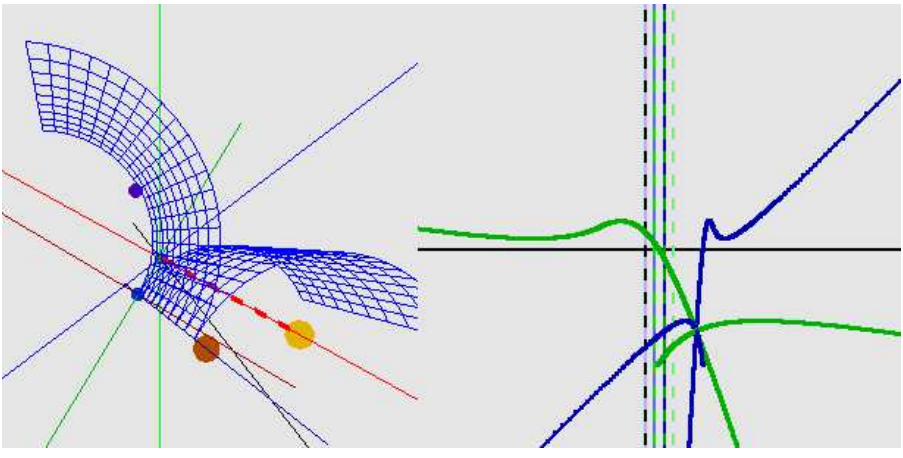


Fig. 19. The two leaves of the surface in the previous figure are extended so that there is a *bitangent* plane, tangent to two rulings. It is represented by two coinciding *crossing points* for $\mathbb{R}^3$ and $N-1$ for $\mathbb{R}^N$. This occurs together with the inflection points in each of the representing curves.

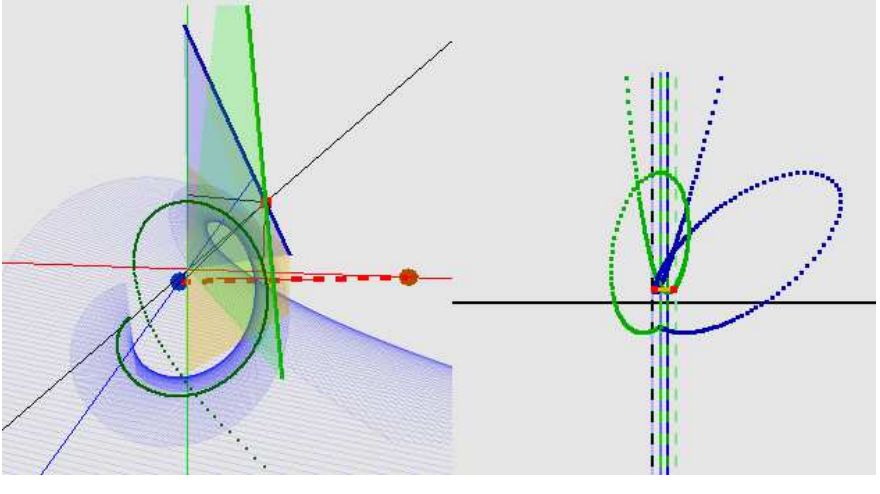


Fig. 20. Developable helicoid and its two representing curves. The two points on the right (one on each curve) represent the tangent plane shown on the left determined by the two line intersections with the first π_0^s, and second π_1^s, super-planes. Again $N-1$ such curves represent a developable helicoid in $\mathbb{R}^N$.

points. The representation is unique if one $(N-1)$-tuple of points of equal y-value is given for matching one set of pieces, for a total of k such tuples. The reconstruction algorithm still works most of the time by producing a maximum

of $O(k^{N-1})$ possible alternative hypersurface *even if the matching tuples are not given a priori*. It fails only when the representing curves have an infinite number of self-tangencies that break the curves into similar pieces, an infinite number of which have the same number of extremal points and the same extremal y-coordinates occurring in exactly the same sequence.

Curve dualities have analogues for developables in $\mathbb{R}^3$ and in higher dimensions as illustrated in Figs. 18, 19. A developable helix and its representation are seen in Fig. 20.

5 Ruled Surfaces

Developable surfaces are a subset of the more general class of *ruled* surfaces $\mathcal{R}$ which are generated by a one parameter family of lines. A ruled surface can be created by the motion of a line called *ruling*. Let $C : \mathbf{y} = \mathbf{y}(s)$ be a curve and $\mathbf{g}(s)$ a unit vector in the direction of a ruling passing through a point $\mathbf{y}(s)$ of C. A point $P : \mathbf{x}(s, v)$ on the surface is then given by

$$\mathbf{x}(s, v) = \mathbf{y}(s) + v\mathbf{g}(s) \quad G(\mathbf{y}', v, \mathbf{g}') = (\mathbf{y}' + v\mathbf{g}') \times \mathbf{g} \neq 0 \quad \forall\ (s, v)\ , \tag{18}$$

The curve $C : \mathbf{y} = \mathbf{y}(s)$ is called the *base curve* or *directix*. The rulings are the curves $s = constant$. Ruled surfaces used as architectural elements are not only beautiful but also have great structural strength. There is a wonderful collection of ruled many other kinds of surfaces in [11].

With $G(\mathbf{y}', \mathbf{v}, \mathbf{g}') = 0$ eq. (18) describes developables which are cones for $\mathbf{y}$ constant and cylinders for $\mathbf{g}(s)$ constant. Allowing $G(\mathbf{y}', \mathbf{v}, \mathbf{g}') \neq 0$, permits ruled surfaces to twist unlike developables. The tangent plane π_P at a point P on a ruled surface ρ contains the whole ruling r_P on P. For another point $Q \in \rho$ the tangent plane π_Q still contains r_p. Whereas all points on a ruling of a developable have the *same* tangent plane, moving along points on a ruling r of a ruled surface causes the corresponding tangent planes to *rotate* about r

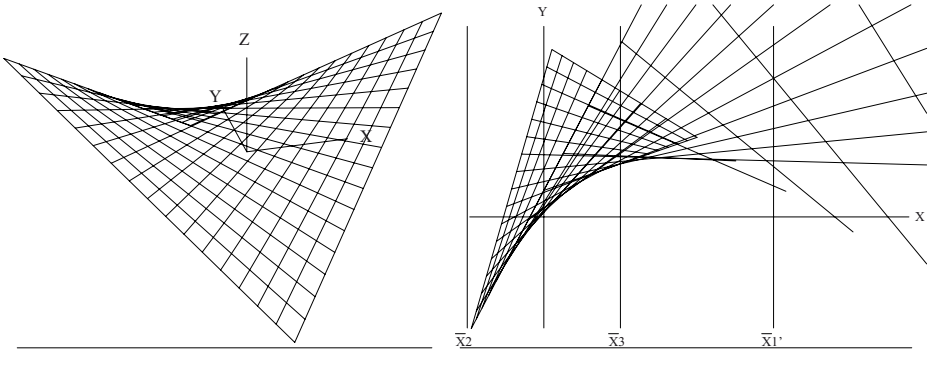


Fig. 21. The saddle (left) is a doubly-ruled surface. One of the two regions representing it on the left. Note the conic (parabolic) boundary.

this being the essential difference and the basis for the ∥-coords representation of ruled surfaces.

A surface can be *doubly-ruled* in the sense that any one of its points is on *two* lines completely contained in the surface (i.e. it can be generated by either one of two sets of moving lines). An elegant result ([12] p. 42) is that in $\mathbb{R}^3$ the *only* doubly ruled surfaces are the quadrics: hyperboloid of one sheet and the saddle (hyperbolic paraboloid).

The representing curve $\bar{\sigma}_{0'}$ of a developable σ can be obtained as the *envelope* of the family of lines $\bar{R}_{0'}$ on the points $\bar{r}_{12}, \bar{r}_{13}, \bar{r}_{23}$ for each ruling r of σ. Similarly $\bar{\sigma}_{1'}$ is the envelope of the lines $\bar{R}_{1'}$ on $\bar{r}_{1'2}, \bar{r}_{1'3}, \bar{r}_{23}$. A matched pair of points $\bar{\pi}_{0'} \in \bar{\sigma}_{0'}$, $\bar{\pi}_{1'} \in \bar{\sigma}_{1'}$ represents the *single* plane containing $r \subset \sigma$ and tangent to everyone of its points. By contrast, a plane tangent to a ruled surface ρ at a point contains a full ruling r but as the point of tangency is translated continuously along r the tangent plane continuously rotates about r. In ∥-coords the points $\bar{\pi}_{0'}, \bar{\pi}_1$ continuously *translate* in tandem (with the same y coordinate) along the corresponding lines $\bar{R}_{0'}, \bar{R}_{1'}$ representing the rotating tangent planes along r – see Fig. 11. An example is the *saddle* Fig. 21 (left) which, since it is a quadric, is representing regions with conic boundaries. The first region and the lines $\bar{R}_{0'}$ tangent to the parabolic boundary (their envelope) is shown on the right.

Theorem 1. *Representation of Ruled Surfaces – A. A ruled surface ρ is represented by the regions $\bar{\rho}_{j'}$, $j = 0, 1$ containing the families of lines $\mathcal{R}_j = \{\bar{R}_j\}$ whose envelopes are the boundaries $\partial\bar{\rho}_{j'}$ specified by Lemma 1.*

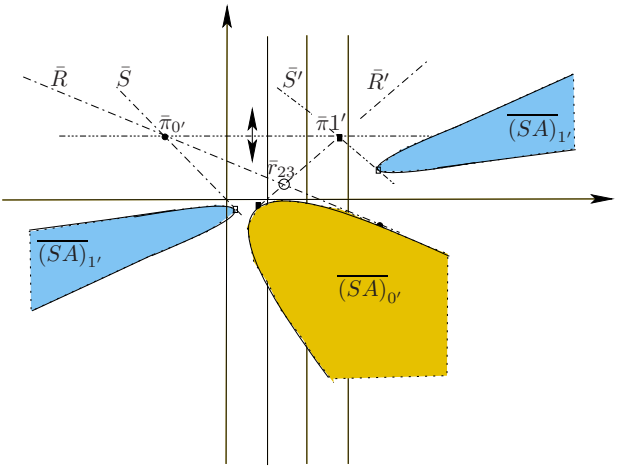


Fig. 22. The saddle SA is represented by the complements of the two shaded regions ($\overline{(SA)}_{0'}$ and $\overline{(SA)}_{1'}$) having parabolic and hyperbolic boundaries respectively. The points $\bar{\pi}_{0'}, \bar{\pi}_{1'}$ representing a tangent plane and the ruling $r \subset \pi$ are constructed with the indicated matching algorithm. A different ruling on π can be constructed since SA is doubly-ruled.

To emphasize, a developable is represented by two curves which are the envelopes of the families of lines $\mathcal{R}_{j'}$ (formed from the representation of its rulings as described above). A ruled surface ρ is represented by two regions $\bar{\rho}_{j'}\,, j = 0, 1$ whose boundaries $\partial\bar{\rho}_{j'}$ are also the envelopes of the lines $\mathcal{R}_{j'}$ (obtained in the same way from the representation of its rulings), **together** with the lines $\mathcal{R}_{j'}$. Equivalently, it is helpful to consider $\partial\bar{\rho}_{j'}$ as a *line-curve* and $\bar{\rho}_{j'}$ the region covered by its tangents. The region's structure enables the construction of the matching algorithm for choosing pairs of points, representing the surface's tangent planes. By the way, the boundary curves deriving from the rulings of a developable *differ* from those obtained from the rulings of a ruled surface which do not necessarily have the *same y range*. These details are clarified with Fig. 22 showing the saddle's full representation.

A famous ruled surface is the *Möbius strip* described by:

$$\begin{gathered}\mathbf{x} = \mathbf{y}(\theta) + v\mathbf{g}(\theta)\ ,\ -\tfrac{1}{2} < v < \tfrac{1}{2}\ ,\ 0 \le \theta \le 2\pi\ , \\ \mathbf{y}(\theta) = (cos\,\theta)\hat{\mathbf{e}}_\mathbf{1} + (sin\,\theta)\hat{\mathbf{e}}_\mathbf{2}\ , \\ \mathbf{g}(\theta) = (sin\,\tfrac{1}{2}\,\theta\,cos\,\theta)\hat{\mathbf{e}}_\mathbf{1} + (sin\,\tfrac{1}{2}\,\theta\,sin\,\theta)\hat{\mathbf{e}}_\mathbf{2} + (cos\,\tfrac{1}{2}\,\theta)\hat{\mathbf{e}}_\mathbf{3}\ ,\end{gathered} \tag{19}$$

It is *non-orientable* in the sense that tracing the normal vector at a moving point along a loop, specified by its directrix the circle $\mathbf{y}(\theta) = cos\,\theta\hat{\mathbf{e}}_\mathbf{1} + sin\,\theta\hat{\mathbf{e}}_\mathbf{2}$, the normal flips by 180^0 from its original direction when the traversal is completed. The strip has only one side, its structure is elucidated in Fig. 24 (right) showing a ruling moving along the directrix and twisting at an angle $\theta/2$ from the vertical, where θ is the angle swept along the directrix, intersecting the central axis in 3 positions inverting its orientation by 180^0 by the time it completes a circuit along the twisted wonder we see in Fig. 24 (left). How does this

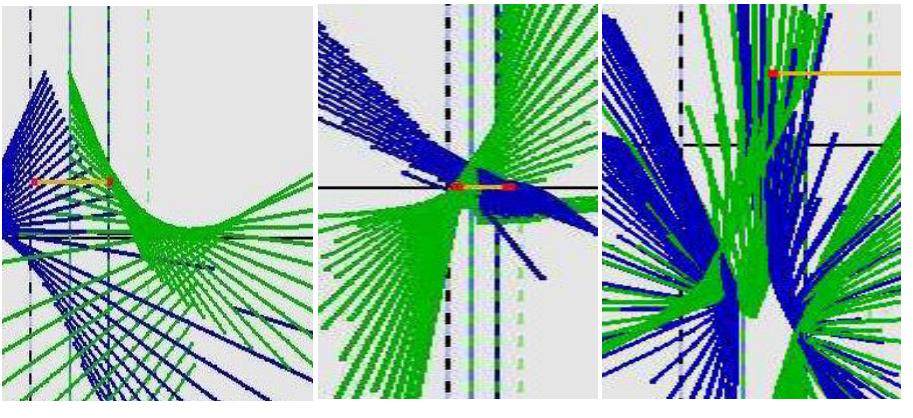


Fig. 23. Representations of various ruled surfaces. The straight lines reveal that the surfaces are ruled and partially outline the regions' boundaries. A tangent plane is represented by the two linked points.

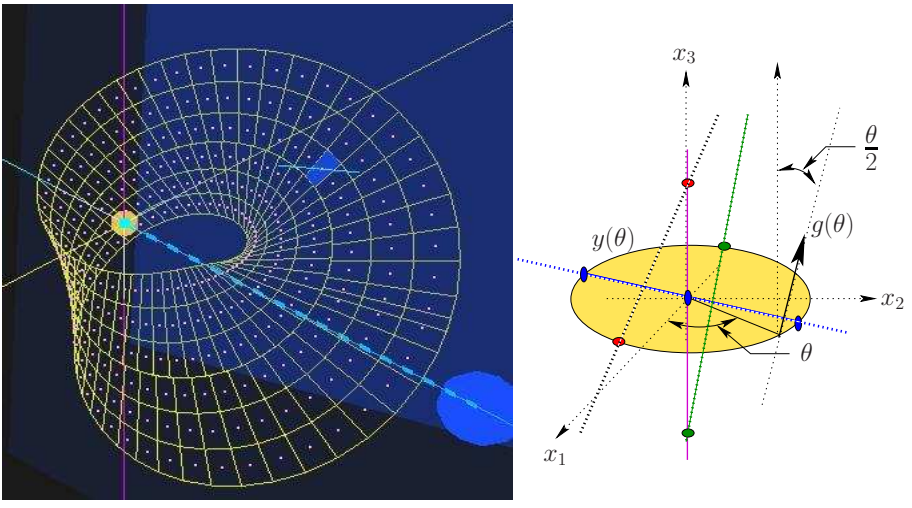


Fig. 24. *Möbius strip* surface and its structure(left) A ruling traversing the circular directrix intersects the central axis three times in one complete circuit

surface appear in ‖-coords? The straight lines, characteristic of ruled surfaces, show up producing dazzling patterns for various orientations Fig. 25. Of particular interest is the search for a pattern characterizing non-orientability. In Fig. 26 (right) is the image of a closed circuit on the strip; a pair of curves with

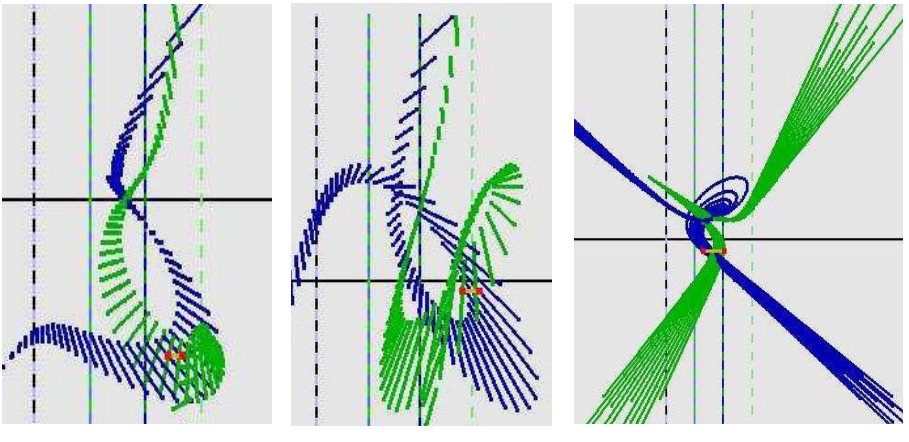


Fig. 25. Representation of a *Mobius strip*. For two orientations the straight lines (left and center) show that it is a ruled surface. On the right the, cusp (being the dual of inflection point) of the "bird-like" pattern may represent the non-orientable twisting resembling a "3-D inflection point". Two linked points represent a tangent plane.

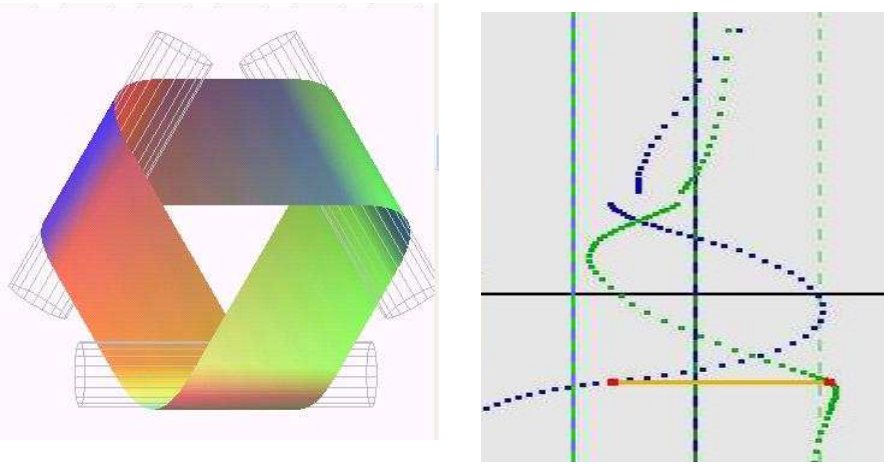


Fig. 26. Visualizing non-orientability. (left) *Möbius strip* (thanks and acknowledgment to [11]) placed on one of the principal planes say x_2x_3. Note the 3 folds (which do need to be symmetrical) resulting in 3 tangents planes perpendicular to the x_2x_3 plane. (right) Traversal of a closed circuit on the *Möbius strip* represented in ∥-coords by two curves having exactly 3 crossing points each corresponding to a tangent plane perpendicular to the x_2x_3 plane.

3 intersections. Each intersection point represents a tangent plane π for which $\bar{\pi}_{0'}$ and $\bar{\pi}_{1'}$ coincide, that is π is perpendicular to the x_2x_3 principal plane. A nice way to understand the twisting is suggested in Fig. 26 (left) where the strip is placed on a plane, in this case the x_2x_3, creating three "folds" which do not have be symmetrically positioned as the ones shown. At each of the folds the tangent plane is perpendicular *and flips* with respect to the x_2x_3 plane – review again the flipping associated with the traversal of a crossing point Fig. 11 where one coefficient changes sign. After traversing all 3 crossing points the tangent plane's orientation differs by 180^0 (all three coefficients changed sign) from that at the start of the circuit. Returning to the representing curves, there are exactly three intersection points corresponding to tangent planes normal to the x_2x_3 plane and these must *represent the tangent planes at each of the three folds*. Crossing the three intersections there is the ensuing flip associated with the strip's non-orientability. For a closed circuit on a doubly-twisted Möbius strip, the representing curves the $\bar{\pi}_{0'}$ and $\bar{\pi}_{1'}$ coincide at six intersection points corresponding to six folds and so on. It appears then that a pattern characteristic of non-orientability has been found whose verification relies on interactively tracing a closed circuit on the representing curves. The picture, of course, is a not proof but may provide insight leading to a proof. If true the generalization to $\mathbb{R}^N$ is as before with $N-1$ such curves with N crossing points corresponding to the N coefficients of a tangent hyperplane and pN crossing points for an N dimensional Möbius strip with p twists.

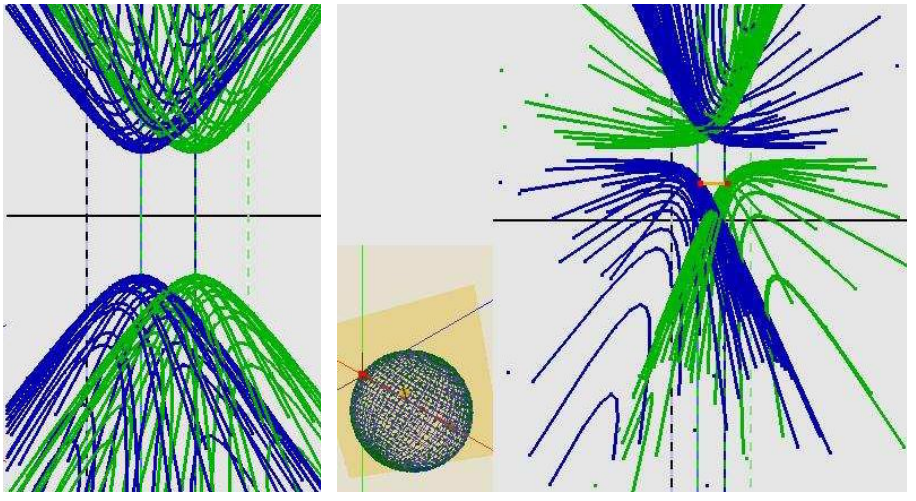


Fig. 27. Representation of a sphere centered at the origin (left) and after a translation along the x_1 axis (right) causing the two hyperbolas to rotate in opposite directions. As in previous figures the two linked points represent a tangent plane.

Here is another thought: twisting a curve about its tangent at a point creates an inflection point. So are the cusps seen in Fig. 25 (right), which are the dual of inflection points, together with the surrounding pattern in the second curve an inkling of an non-orientable twist; like a "3-D inflection point" produced by twisting and then joining a curved strip? Though imprecise this depiction, in the spirit of visualization, is intuitive stimulating ideas. Does the bird-like pattern represent the Möbius strip's non-orientability? We sum up the representation results thus far.

Theorem 2. (Representation of developable and ruled surfaces in $\mathbb{R}^N$) *A surface $\sigma \subset \mathbb{R}^N$ and represented by $N-1$ regions $\bar{\sigma} \subset \mathbb{P}^2$ is*

1. **ruled but not developable** $\Leftrightarrow$ *all points of $\bar{\sigma}$ are* **exterior** *with respect to the boundary of its $N-1$ regions,*
2. **developable** $\Leftrightarrow$ *$\bar{\sigma}$ consists of $N-1$ curves, the boundaries of $\bar{\sigma}$, all having the same y-range,*

6 More General Surfaces

A sphere in $\mathbb{R}^3$ is represented by two hyperbolas as seen in Fig. 27 consisting of two hyperbolas. Due to their symmetry the hypercube, as in Fig. 5 for $\mathbb{R}^5$ and a hypersphere in $\mathbb{R}^N$ are represented by $N-1$ identical patterns. Unlike ruled surfaces, here the representing points are *interior* to the hyperbola; this is an

easy consequence of convexity. Ellipsoids and paraboloids of one sheet are also represented by hyperbolas. A bounded convex surface (abbr. by *bcs*) surface in $\mathbb{R}^3$ has a pair of tangent (or supporting planes) planes in any direction including the direction of u; the line through $(0,0,0)$ and $(1,1,1)$ and according to corollary 2 the boundaries of each of its representing regions has two asymptotes resembling hyperbolas as in Fig. 28 (left). For easy reference such curves are called *generalized hyperbolas* and abbreviated by *gh*. Similarly, a bounded convex hypersurface in $\mathbb{R}^N$ is represented by $N-1$ *gh* regions each with a pair of asymptotes. Conversely, consider a bounded hypersurface whose representation at a specific orientation consists only of *gh* regions. By the the *translation* $\leftrightarrow$ *rotation* duality, evident in Fig. 27, the regions remain *gh* for *all* orientations. The representation of an unbounded convex hypersurface (abbr. *ucs*) also consists of *gh* regions unless it has some (say m) ideal points in the direction of u in which case m of its representing regions are parabola-like (i.e. have one ideal point – called *generalized-parabola* and abbr. *gp*) and $N-m-1$ are *gh*. Without entering into details we conjecture that the *gp* regions have vertical ideal points. This yields a powerful result, important in many applications, enabling the detection and viewing of convexity in $\mathbb{R}^N$.

Theorem 3. (Convexity) *A hypersurface $\sigma \subset \mathbb{R}^N$ is bounded and convex $\Leftrightarrow$ for an orientation it is represented by $N-1$* gh *regions. If a representation has at least one* gp *region and the rest are* gh *regions then σ is unbounded and convex.*

Whereas the interior points of the hyperbolic regions properly matched represent the sphere's tangent planes there is another interesting matching which gives a feel about the sphere's interior. Consider the representation of concentric spheres (centered at the origin) with radii r ranging between a minimum r_m and r_M

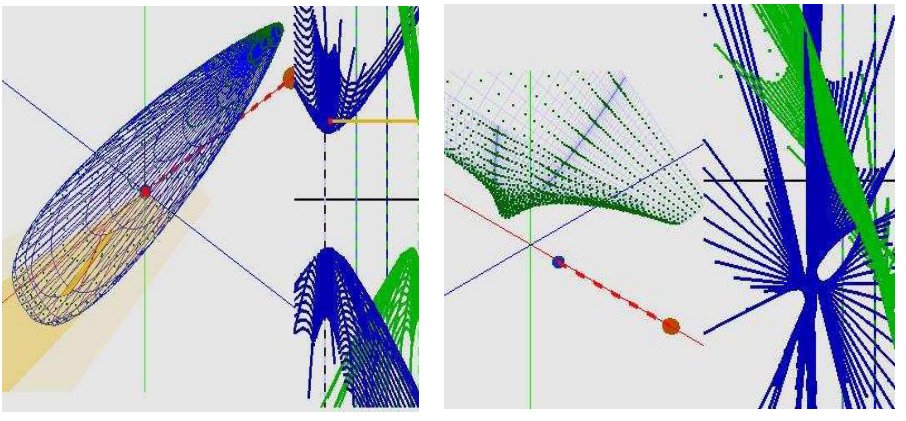


Fig. 28. Representation of a non-quadric convex surface by *gh* regions (left) and a non-convex one with a bump (right)

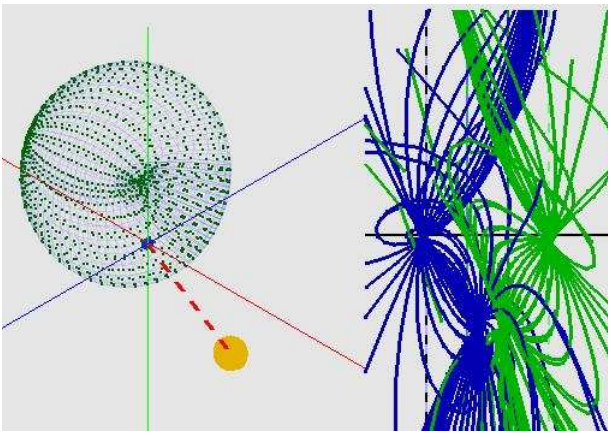


Fig. 29. A non-convex surface with 3 dimples (depressions) corresponding to the swirls in the representation (right)

and maximum values. As r increases the hyperbolas' vertices and the rest of the curve rise linearly with r enlarging the region in between the upper and lower branches. Let a pair of points at $y = y_a$ be given on the hyperbolic boundaries representing a tangent plane π_a of one of the spheres. Its radius a is found from the vertices of the hyperbolas, the points are on, which are at $y = \pm\sqrt{2/3}a$. Draw the vertical lines on the pair of given points. Any other pair of points at

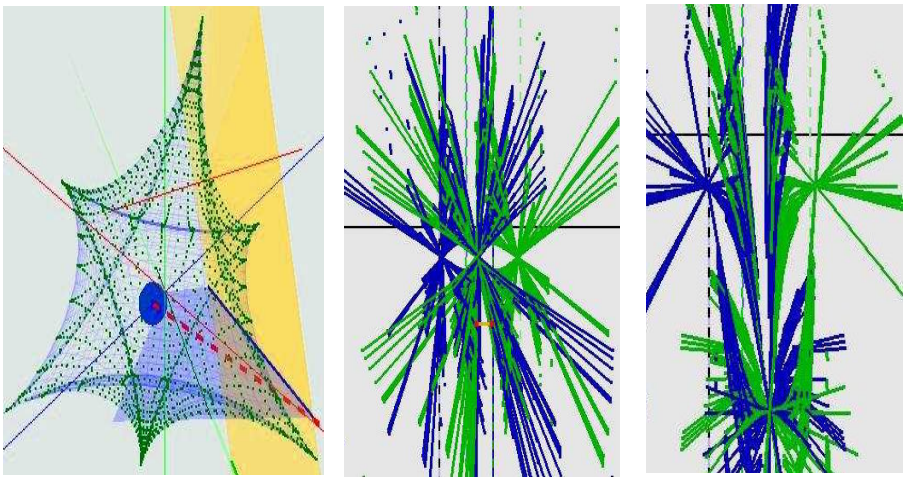


Fig. 30. Astroidal surface and representations for two orientations. The 3 crossing-points pattern persists as the orientation is changed.

$y = y_c$ represents a tangent plane π_c parallel to π_a on the sphere with radius $c = y_c/y_a a$. A similar analysis for any bounded convex hypersurface reveals that as it expands the distance between the branches of the *gh* representing regions increases.

It is instructive to examine the representation of surfaces having small deviations – "perturbations" – from convexity. In Fig. 28 on the right is a surface σ_b with a small bump. Its orientation was interactively varied in every imaginable way and significantly its representations like the one shown were invariably markedly different than a *gh*. Even for the orientations where the intersections of σ_b with the super-planes, see Corollary 2, are bounded and convex the representing regions are not *gh*. No less significant is that the representing regions are not chaotic having very interesting shapes continuously varying with changing orientation. Proceeding, a surface σ_d with three small dimples – depression being considered here as a perturbation of convexity "opposite" to that in σ_b – shown in Fig. 29 is explored next. Interesting representation patterns with a "swirl" corresponding to each dimple are seen with varying orientations. Again it is rather remarkable that for a multitude of orientations none of the representations consists of *just* two hyperbolas as may naively be expected.when the intersection with the super-planes is a bounded convex set. There seem to be "invariants" like the "swirls" $\leftrightarrow$ "dimples" worth further investigation.

The representation of the asteroidal surface [11] in Fig. 30 has a pattern with 3 crossing points which persists in several orientations. Upon careful scrutiny only at the mid-point do curves from the two regions intersect. Could this point represent a tangent plane approaching one of the surface's vertices and becoming a bitangent plane in the limit? Another association that comes to mind is of an exterior supporting plane touching the surface at exactly three vertices. The significant finding from these examples is that, the ||-coords representation even of complex non-convex surfaces in various orientations is not chaotic but consists of discernible patterns corresponding to their properties which may also lead to new surface classifications. So based on this discussion is what happened to the claim the representing boundaries are the images of the surface's intersection with the super-planes? This question is connected with some deeper aspects in the theory of envelopes [13]. Briefly, the envelope of a family of planar curves is a "reasonable" curve as long as the *curves are all on the same side of envelope*. These correspond to the boundary curves prescribed by Lemma 2.

Families of Proximate Planes and Proximate Developables. A central problem in many applications e.g. Statistics(Regression), Geometric Modeling (Tolerancing), Computer Vision (Picture Reconstruction) etc. is concerned with "proximity" of planes. Let a a family of *proximate* planes be generated by

$$\begin{aligned} \Pi = \Pi : \{c_1 x_1 + c_2 x_2 + c_3 x_3 = c_0, \\ c_i \in [c_i^-, c_i^+], \quad i = 0, 1, 2, 3\}. \end{aligned} \tag{20}$$

In Fig. 31 (left) we see the two point clusters generated by randomly choosing planes in Π and plotting their two point representation. Closeness is apparent

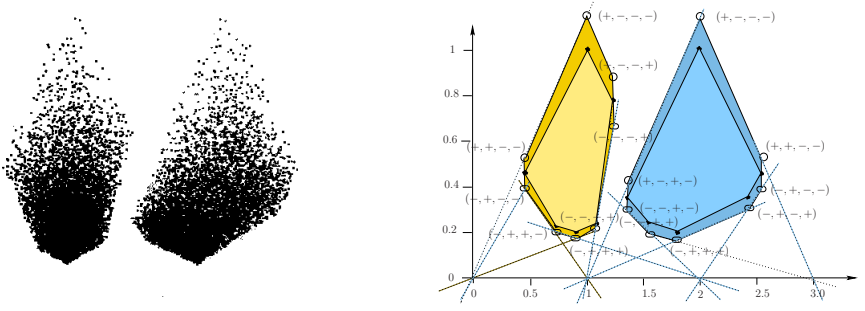


Fig. 31. (Left) Pair of point clusters representing close planes.(Right) The hexagonal regions (interior) are the regions containing the points $\bar{\pi}_{123}$ (left) and $\bar{\pi}_{1'23}$ for the family of planes with $c_0 = 1$. For variable c_0 the regions (exterior) are octagonal with two vertical edges.

and the distribution of the points is not chaotic. The precise polygonal patterns are shown on the right. Not only it is possible to see near coplanarity but also see, estimate and compare errors. Families of proximate hyperplanes in $\mathbb{R}^N$ are represented by $N-1$ convex polygons having $2(N+1)$ edges constructed by a nice algorithm with $O(N)$ computational complexity [14]. The family Π forms a non-convex difficult to visualize surface in $\mathbb{R}^3$ yet its image in ∥-coords consists of two convex octagons which are easy to work with. By dividing the polygons into triangles a point in one polygon can be matched to another in the second which represent a valid plane in Π. This generalizes completely to $\mathbb{R}^N$ and prompts another idea.

In our surroundings there are lots of objects that are "nearly" developable and many parts are manufactured by deforming flat sheets. This and the simple ∥-coordinate representation motivate the study of "approximate developables" as for the proximate planes above, by introducing perturbations in the developables' equation allowing $|(\mathbf{y}' + v\mathbf{g}') \times \mathbf{g}| = \epsilon \neq 0$ and there may be other useful definitions. The conjecture is that families of "nearly developables" are represented in ∥-coords by curved "strips" with properties "close" to those discovered by the duality (of developables) with space curves and amenable to an intuitive treatment. New classes of surfaces (patches), with their N-D generalizations, easy to visualize and work with interactively may emerge suitable for applications like Geometric Modeling.

Engineering drawings depict a 3-D object in terms of 2 principal projections which are pointwise *linked* i.e. 2 linked regions. In ∥-coords an object is not represented by its points but by its tangent planes. Unlike a point, the tangent plane at the point also provides *local directional information* like the normal direction. The ∥-coords representation depicts the object in terms of 2 linked regions formed by its tangent planes containing much more information than the point projections. Cavities, like the dimples in Fig. 29 are hidden in a projection but can not hide in the ∥-coords representation. This and other properties like

non-orientability and proximity are revealed leading to their higher dimensional generalizations. We rest our case for considering, working, studying and enjoying this surface representation.

Acknowledgements

We are grateful to David Adjiashvili who wrote the software constructing the representation of many surfaces presented here and to the referee for valuable suggestions.

References

1. Inselberg, A.: Don't panic ... do it in parallel! Comput. Statit. 14, 53–77 (1999)
2. Inselberg, A.: Parallel Coordinates: VISUAL Multidimensional Geometry and its Applications. Springer, Heidelberg (2007)
3. Inselberg, A., Dimsdale, B.: Multidimensional lines ii: Proximity and applications. SIAM J. of Applied Math. 54(2), 578–596 (1994)
4. Inselberg, A., Dimsdale, B.: Multidimensional lines i: Representation. SIAM J. of Applied Math. 54(2), 559–577 (1994)
5. Chatterjee, A.: Visualizing Multidimensional Polytopes and Topologies for Tolerances. Ph.D. Thesis, Dept. Comp. Sci. Univ. of S. Calif (1995)
6. Eickemeyer, J.: Visualizing p-flats in N-space using Parallel Coordinates. Ph.D. Thesis, Dept. Comp. Sc. UCLA (1992)
7. Beardon, A.: The Geometry of Discrete Groups. Springer, Heidelberg (1983)
8. Fischer, G.: Plane Algebraic Curves. Amer. Math. Soc. Press, Providence (2001)
9. Ernstrom, L.: A plucker formula for the singular projective varieties. Commut. Algebra 5(9), 2897–2901 (1997)
10. Ya, V.O.: Some integral calculus based on euler characteristic, lect. notes. Amer. Math. Monthly 1346, 127–138 (1988)
11. Mathcurve.com: Surfaces (2006), `http://www.mathcurve.com/surfaces/surfaces.shtml`
12. Brannan, D.A., Esplen, M., Gray, J.J.: Geometry. Cambridge University Press, New York (1999)
13. Boltyanskii, V.G.: Envelopes, R.B. Brown translator (original in Russian). Pergamon Press, New York (1964)
14. Matskewich, T., Inselberg, A., Bercovier, M.: Approximated Planes in Parallel Coordinates. In: Proc. Geom. Model. Conf. St. Malo, Vanderbilt Univ. Press, pp. 257–266 (2000)

Discrete Surface Ricci Flow: Theory and Applications

Miao Jin, Junho Kim, and Xianfeng David Gu

State University of New York at Stony Brook, Stony Brook, NY, USA
gu@cs.sunysb.edu

Abstract. Conformal geometry is at the core of pure mathematics. Conformal structure is more flexible than Riemaniann metric but more rigid than topology. Conformal geometric methods have played important roles in engineering fields.

This work introduces a theoretically rigorous and practically efficient method for computing Riemannian metrics with prescribed Gaussian curvatures on discrete surfaces—discrete surface Ricci flow, whose continuous counter part has been used in the proof of Poincaré conjecture. Continuous Ricci flow conformally deforms a Riemannian metric on a smooth surface such that the Gaussian curvature evolves like a heat diffusion process. Eventually, the Gaussian curvature becomes constant and the limiting Riemannian metric is conformal to the original one.

In the discrete case, surfaces are represented as piecewise linear triangle meshes. Since the Riemannian metric and the Gaussian curvature are discretized as the edge lengths and the angle deficits, the discrete Ricci flow can be defined as the deformation of edge lengths driven by the discrete curvature. The existence and uniqueness of the solution and the convergence of the flow process are theoretically proven, and numerical algorithms to compute Riemannian metrics with prescribed Gaussian curvatures using discrete Ricci flow are also designed.

Discrete Ricci flow has broad applications in graphics, geometric modeling, and medical imaging, such as surface parameterization, surface matching, manifold splines, and construction of geometric structures on general surfaces.

1 Introduction

Conformal geometry offers rigorous and powerful theoretic tools for practical engineering applications. Ricci flow is a novel curvature flow method in computational conformal geometry, which will play important roles in practice because of its universality and flexibility.

1.1 Conformal Geometry

Conformal geometry is at the core of pure mathematics, which is the intersection of complex analysis, algebraic topology, differential geometry, algebraic geometry, and many other fields in mathematics.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 209–232, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Conformal geometry studies the conformal structures of Riemann surfaces. Conformal structure is a fundamental geometric structure, which is more rigid than topological structure but more flexible than Riemannian metric structure. All surfaces can be deformed to three canonical spaces preserving geometric information. This fact greatly simplifies the theoretic arguments in mathematics. Furthermore, all surfaces can be classified according to their conformal structures. The 'shape space' of conformal equivalent classes form a finite dimensional Riemannian manifold, which offers a universal framework for shape analysis.

With the development of 3D acquisition technologies and computational power, conformal geometry plays more and more important roles in engineering fields. For example, conformal geometry has been broadly applied in computer graphics, computer vision, geometric modeling and medical imaging. The theoretic foundation for computational conformal geometry is developing rapidly and many practical algorithms converted from classical theories in conformal geometry have been invented.

So far, the computational methodologies in conformal geometry for general surfaces are mainly in the following categories: harmonic maps, holomorphic differentials, and newly invented method in geometric analysis, *Ricci flow*. Ricci flow is very powerful and flexible, which will generate great impact in engineering field. In this work, we mainly focus on the introduction to the theories and algorithms of discrete surface Ricci flow.

1.2 Ricci Flow

Recently, the term *Ricci flow* has become popular, due to the fact that it has been applied for the proof of the Poincaré conjecture on 3-manifolds [1,2,3]. Richard Hamilton introduced the Ricci flow for Riemannian manifolds of any dimension in his seminal work [4] in 1982. Intuitively, a surface Ricci flow is the process to deform the Riemannian metric of the surface. The deformation is proportional to Gaussian curvatures, such that the curvature evolves like the heat diffusion.

It has been considered as a powerful tool for computing the conformal Riemannian metrics with prescribed Gaussian curvatures. For many engineering applications, it is also highly desirable to compute Riemannian metrics on surfaces with prescribed Gaussian curvatures, such as parameterization in graphics, spline construction in geometric modeling, conformal brain mapping in medical images, and so on.

Surface parameterization refers to the process of mapping a surface onto a planar domain. If such a parameterization is known, any functions or signals (e.g., texture) on the flat parametric domain can be easily pulled back to the surface, such that complicated processing on surfaces can be transferred to easy computing on the flat parametric domain. Therefore it is a key ingredient for digital geometry processing, such as texturing [5], deformation [6], and resampling [7]. The process for parameterizing surfaces is equivalent to finding a special flat Riemannian metric, with *zero* Gaussian curvatures everywhere.

Constructing splines whose parametric domain is an arbitrarily topological manifold is an important issue for computer-aided geometric modeling [8]. In order to define such parameters and knots of the spline, a special atlas of the surface is required such that all local coordinate transition maps are affine [9]. One way to construct such an atlas is to find a flat metric of the surface first, then locate a collection of patches covering the whole surface, and flatten each patch using the flat metric to form an atlas. Here again, the key step is to obtain the flat metric of the given surface.

In medical imaging field, it is important to deform the human brain cortex surface to the unit sphere in order to easily compare and register several different brain cortexes on a canonical domain [10]. This is equivalent to find a Riemannian metric on the cortex surface, such that the Gaussian curvature induced by the metric equals to *one* everywhere.

Comparing to existing methods, which can only handle a subproblem in the scope of Euclidean parametrization, Ricci flow can handle arbitrary topologies and find arbitrary conformal mappings, which include not just Euclidean, but also hyperbolic and spherical parameterizations.

The discrete Ricci flow on piecewise linear surfaces was introduced in [11]. The existence and convergence of the discrete Ricci flow for surfaces were established. However, the discrete Ricci flow is not a very efficient algorithm for practical use due to the gradient nature of the flow. Recently, we improve the gradient descent method by the Newton's method and drastically speed up the search for the limiting metric by the order of magnitudes. Furthermore, we generalize the results from constant discrete curvature to arbitrarily prescribed discrete curvature, from the metric induced by the combinatorial structure of the mesh to the induced Euclidean metric. An effective and complete system to compute Riemannian metrics with prescribed Gaussian curvatures on generally topological surfaces has been developed in this paper based on discrete Euclidean Ricci flow, discrete hyperbolic Ricci flow, and discrete spherical Ricci flow.

1.3 Outline

In the next Sec. 2, we will briefly review most related works in computer graphics field and discrete complex analysis; in Sec. 3, theoretic backgrounds in differential geometry and Rieman surface are introduced; in Sec. 4, the theories for major computational methodologies are introduced, including harmonic maps, holomorphic differential forms and Ricci flow; in Sec. 5, the algorithm and theories of discrete Ricci flow are thoroughly explained; practical applications are given in Sec. 6; in the conclusion Sec. 7, future directions are pointed out.

2 Previous Work

In computer graphics and discrete mathematics, much sound research has focused on discrete conformal parameterizations. Here, we give a brief overview of related work, and refer readers to [12,13] for thorough surveys.

All parameterizations can be classified according to the type of output produced, which can be *a vector valued function, i.e. a mapping*, *a holomorphic differential form* or *a flat Riemannian metric*. In general, the derivative of a conformal map is a holomorphic 1-form; each holomorphic 1-form induces a flat metric. Therefore, methods which compute metrics are the most general, although they are more expensive to compute.

2.1 Mappings

First order finite element approximations of the Cauchy-Riemann equations were introduced by Levy et al. [5]. Discrete intrinsic parameterization by minimizing Dirichlet energy was introduced by [14], which is equivalent to least-squares conformal mapping [5]. Discrete harmonic maps were computed using the cotan-formula in [15]. Mean value coordinates were introduced in [16]; these generalize the cotan-formula. All these linear methods can easily incorporate free boundary conditions to improve the quality of the parameterization produced, such as the methods in [14] and [17]. Discrete spherical conformal mappings are used in [18] and [10].

2.2 Holomorphic Forms

Holomorphic forms are used in [19] to compute global conformal surface parameterizations for high genus surfaces. Discrete holomorphy was introduced in [20] using discrete exterior calculus [21]. The problem of computing optimal holomorphic 1-forms to reduce area distortion was considered in [22]. Gortler et al. [23] generalized 1-forms to the discrete case, using them to parameterize genus one meshes. Recently, Tong et al. [24] generalized the 1-form method to incorporate cone singularities.

2.3 Metrics

There are three major methods for computing edge lengths (or equivalently the angles): angle based flattening, circle packing, and circle patterns.

Sheffer and Sturler [25] introduced the angle based mesh flattening method. This works by posing a constrained quadratic minimization problem seeking to find corner angles which are close to desired angles in a weighted $L2$ norm. The efficiency and stability of ABF are improved in [26] by using advanced numerical algorithm and hierarchical method.

The circle packing method was introduced in [27]. Continuous conformal mappings can be characterized as mapping infinitesimal circles to infinitesimal circles. Circle packings replace infinitesimal circles with finite circles. In the limit of refinement the continuous conformal maps are recovered [28]. Collins and Stephenson [29] have implemented these ideas in their software CirclePack.

The first variational principle for circle packings, was presented in a seminal paper by Colin de Verdière [30]. Circle patterns based on those in Bobenko and Springborn [31] have been applied for parameterization in [32]. Springborn [33] shows that in theory, circle packing and circle patterns are equivalent.

2.4 Ricci Flow

Recently, a novel curvature flow method in geometric analysis is introduced to prove the Poincaré conjecture, the *Ricci Flow*. Ricci flow refers to conformally deform the Riemannian metric of a surface by its Gaussian curvature, such that the curvature evolves according to a heat diffusion process. Ricci flow is a powerful tool to compute the Riemannian metric by the curvature. It can be applied for discrete conformal parameterizations.

The connection between circle packing and smooth surface Ricci flow [34] was discovered in [11]. Conventional circle packing only considers combinatorics. The discrete Ricci flow method was introduced in [35,36], which incorporate geometric information and was applied for computing hyperbolic and projective structure and manifold splines.

3 Theoretical Background

In this section, we briefly introduce the major concepts in differential geometry and Riemann surface theory, which are necessary to explain the theory of the Ricci flow. We refer readers to [37,38] for detailed information.

3.1 Riemannian Metric

Suppose $D \subset \mathbb{R}^2$ is a planar domain, a map $\mathbf{r} : D \to \mathbb{R}^3$ is with at least C^2 continuity. Let $\mathbf{r}_i = \partial \mathbf{r}/\partial u_i, i = 1, 2$ be the tangent vectors along the iso-parametric curves. If

$$\mathbf{r}_1 \times \mathbf{r}_2 \neq 0,$$

then $\mathbf{r}$ is called a *regular surface*. The *normal* is defined as

$$\mathbf{n} = \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|},$$

the map $G : \mathbf{r}(u_1, u_2) \to \mathbf{n}(u_1, u_2) \in \mathbb{S}^2$ is called the *Gauss map*, where $\mathbb{S}^2$ is the unit sphere. The length of a general tangent vector $d\mathbf{r} = \mathbf{r}_1 du_1 + \mathbf{r}_2 du_2$ can be computed

$$ds^2 = < d\mathbf{r}, d\mathbf{r} > = (du_1 du_2) \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \begin{pmatrix} du_1 \\ du_2 \end{pmatrix}$$

where $<,>$ is the inner product in $\mathbb{R}^3$, and $g_{ij} = < r_i, r_j >$. The matrix $\mathbf{g} = (g_{ij})$ is called the *Riemannian metric tensor*, which defines the inner products $<,>_{\mathbf{g}}$ on the tangent planes of the surface. The angles between two tangent vectors can be measured by $\mathbf{g}$. Suppose $\delta\mathbf{r} = \mathbf{r}_1 \delta u_1 + \mathbf{r}_2 \delta u_2$ is another tangent vector, the angle between $d\mathbf{r}$ and $\delta\mathbf{r}$ measured by $\mathbf{g}$ is $\theta_{\mathbf{g}}$, then

$$\theta_{\mathbf{g}} = \cos^{-1} \frac{< d\mathbf{r}, \delta\mathbf{r} >_{\mathbf{g}}}{\sqrt{< d\mathbf{r}, d\mathbf{r} >_{\mathbf{g}}} \sqrt{< \delta\mathbf{r}, \delta\mathbf{r} >_{\mathbf{g}}}} . \tag{1}$$

Suppose $\lambda : (u_1, u_2) \rightarrow \mathbb{R}$ is a real function defined on the surface, then $\bar{\mathbf{g}} = e^{2\lambda(u_1,u_2)}\mathbf{g}$ is another Riemannian metric,

$$< d\mathbf{r}, \delta\mathbf{r} >_{\bar{\mathbf{g}}} = e^{2\lambda} < d\mathbf{r}, \delta\mathbf{r} >_{\mathbf{g}} . \tag{2}$$

The angle between $d\mathbf{r}$ and $\delta\mathbf{r}$ measured by $\bar{\mathbf{g}}$ is $\theta_{\bar{\mathbf{g}}}$. Plugging Eqn. 2 to 1 directly, we obtain

$$\theta_{\mathbf{g}} = \theta_{\bar{\mathbf{g}}}.$$

Namely, the angles measured by $\mathbf{g}$ equal to those measured by $\bar{\mathbf{g}}$. Therefore, we say $\bar{\mathbf{g}}$ and $\mathbf{g}$ are *conformally equivalent*. 'conformal' means the angle preserving, while the area distorted, with $e^{2\lambda}$ measuring the area distortion and called the *conformal factor*.

All Riemannian metrics are surfaces are classified by the conformal equivalence relation. Each conformal equivalent class is called a *conformal structure*. Any Riemannian metric on the surface are locally conformal equivalent to the Euclidean flat metric. Namely, one can choose a special parameters, and the change of parameterization doesn't affect the metric, such that the metric is represented as

$$ds^2 = e^{2\lambda}(du_1^2 + du_2^2). \tag{3}$$

Such kind of parameterizations are called the *isothermal coordinates* of the surface.

3.2 Gaussian Curvature

The Gauss map maps the surface to the unit sphere. The derivative map of the Gauss map is the first order linear approximation to it. The Jacobin of the derivative map measures the local area distortion between the surface and the image on the sphere, which is called the *Gaussian curvature* K. Under the isothermal coordinates Eqn. 3, the Gaussian curvature is represented as

$$K = -e^{-2\lambda}\Delta\lambda,$$

where Δ is the *Laplace operator*,

$$\Delta = \frac{\partial^2}{\partial u_1^2} + \frac{\partial^2}{\partial u_2^2}.$$

$e^{-2\lambda}\Delta$ is called the *Laplace-Beltrami operator*.

Along the boundary of the surface, the tangential component of the curvature of the boundary curve is defined as the *geodesic curvature* k. Under the isothermal coordinates, the geodesic curvature is represented as

$$k = -e^{-\lambda}\partial_{\mathbf{n}}\lambda,$$

where $\mathbf{n}$ is the exterior tangent vector perpendicular to the boundary.

The total curvature of a surface is solely determined by the topology of the surface and independent of the metric. Gauss-Bonnet theorem [37] explains

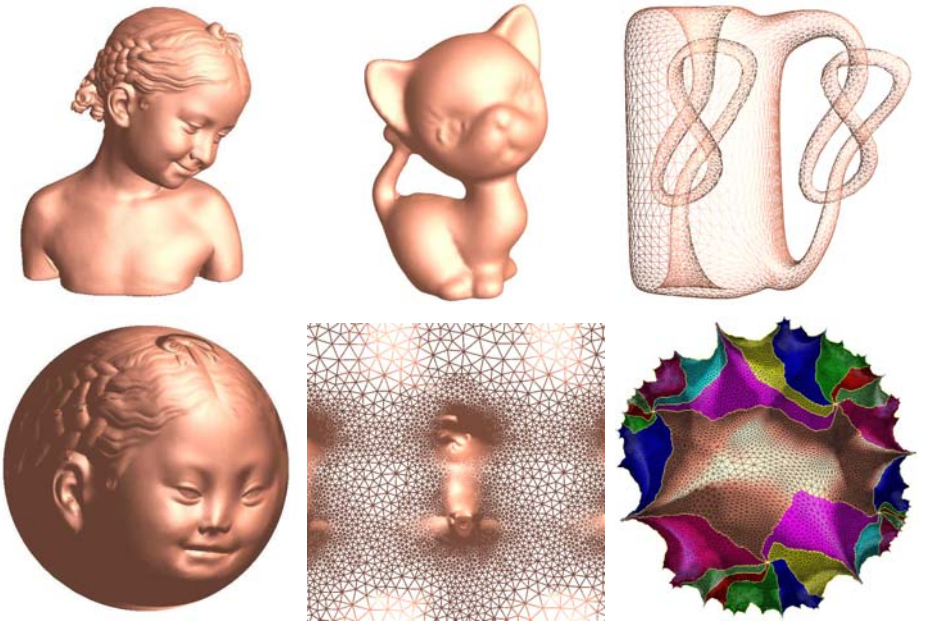


Fig. 1. Uniformization Theorem: all surfaces with Riemannian metric can be conformally embedded onto three canonical spaces: sphere, plane and hyperbolic space

the connection between the total Gaussian curvatures on a surface S and the topology of S:

$$\int_S K dA + \int_{\partial S} k_g ds = 2\pi\chi(S), \tag{4}$$

where $\chi(S)$ is the Euler number of S, $\chi(S) = 2 - 2g - b$, g, b are the genus and the number of the boundaries of the surface.

3.3 Conformal Metric Deformation

Under a conformal metric $\bar{\mathbf{g}}$, the curvatures are changed accordingly. The following *Yamabi equations* describe the relation between the conformal metric deformation and the curvature change, suppose $\bar{\mathbf{g}} = e^{2\lambda}\mathbf{g}$, then

$$\bar{K} = e^{-2\lambda}(K - \Delta\lambda), \bar{k} = e^{-\lambda}(k - \partial_{\mathbf{n}}\lambda), \tag{5}$$

where Δ is the Laplace-Beltrami operator under the metric $\mathbf{g}$.

It has the fundamental importance to solve the Riemannian metric $\bar{\mathbf{g}}$ from the target curvature $\bar{K}$ and $\bar{k}$. In each conformal equivalent class, there exists a unique metric, such that the induced curvature is one of the three constants everywhere $\{+1, 0, -1\}$. This fact is formulated as the Uniformization theorem [27].

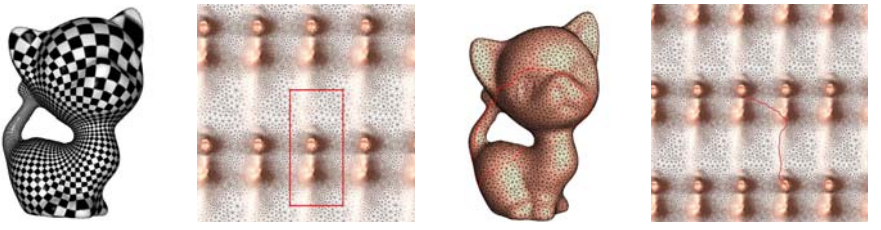


Fig. 2. (a) The kitten surface with the flat metric computed using Ricci flow method. (b) Its universal covering space and one fundamental domain (the rectangle). (c) A loop on surface. (d) The loop lifted to a path in UCS.

Theorem 1 (Uniformization Theorem). *Let* $(S, \mathbf{g})$ *be a compact two-dimensional Riemannian manifold with Riemannian metric* $\mathbf{g}$*, then there is a unique metric* $\bar{\mathbf{g}}$ *conformal to* $\mathbf{g}$ *with constant Gaussian curvatures. The constant value is one of the* $\{+1, 0, -1\}$.

Such metric is called the *uniformization metric*. According to Gauss-Bonnet theorem (Eqn. 4), the sign of the constant Gaussian curvatures is determined by the Euler number of the surface. Therefore, any closed 2-manifold surface can be locally isometrically embedded onto one of the three canonical surfaces with respect to its Euler number χ; the *sphere* $\mathbb{S}^2$ for genus zero surfaces with $\chi > 0$, the *plane* $\mathbb{E}^2$ for genus one surfaces with $\chi = 0$, and the *hyperbolic space* $\mathbb{H}^2$ for high genus surfaces with $\chi < 0$ (see Fig. 1).

3.4 Universal Covering Space

Suppose S is a surface, then $(\bar{S}, \pi)$ is called the *universal covering* space of S, if $\bar{S}$ is simply connected and locally $\pi : \bar{S} \to S$ is bijective and continuous. Fig. 2(b) shows the universal covering space of the genus one kitten surface (a). A *deck transformation* is an automorphism $\tau : \bar{S} \to \bar{S}$, such that $\pi \circ \tau = \pi$ as shown in Fig. 3. All deck transformations form a group, called the *deck transformation group* and denoted as $\mathrm{Deck}(S)$. Suppose $D \subset \bar{S}$ is simply connected, for arbitrary 2 deck transformations τ_1, τ_2, $\tau_1(D) \cap \tau_2(D) = \emptyset$, $\cup_{\tau \in \mathrm{Deck}(S)} \tau(D) = \bar{S}$, then D is called a *fundamental domain* of S. The rectangles in Fig. 3 illustrate two fundamental domains.

Suppose a point $p \in S$ on the surface, the pre-images of p on the universal covering space are $\pi^{-1}(p) = \{\bar{p}_0, \bar{p}_1, \bar{p}_2, \cdots\} \subset \bar{S}$, then $\pi^{-1}(p)$ has a one to one map to the deck transformation group $\mathrm{Deck}(S)$. For any $\bar{p}_k \in \pi^{-1}(p)$, there exists a unique deck transformation $\tau_k \in \mathrm{Deck}(S)$, such that $\tau_k(\bar{p}_0) = \bar{p}_k$: see Fig. 3.

The deck transformations of $\bar{S}$ with the uniformization metric are rigid motions in the canonical spaces, which form a group (the *Fuchsian group* in hyperbolic space). Such a group can represent the conformal structure of the surface S.

The universal covering space of a surface with negative Euler number can be embedded isometrically onto the hyperbolic space with the uniformization

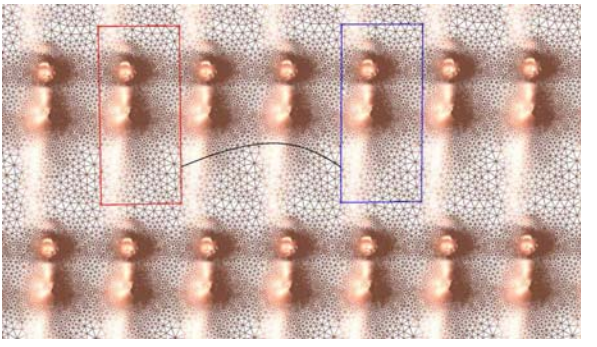


Fig. 3. The universial covering space of the bunny surface. A deck transformation maps the left foundamental domain to the right.

metric. The common hyperbolic space models include Poincaré disk and Klein model. The Poincaré disk is the unit disk on the complex plane, $|z| < 1$ with the metric $ds^2 = dz\, d\bar{z}/(1 - \bar{z}z)$. The geodesics are circular arcs perpendicular to the boundary. The Klein model is also the unit disk. The map from the Poincaré model to the Klein model is

$$\beta(z) = \frac{2z}{1 + z\bar{z}}.$$

3.5 Riemann Surface

A Riemann surface is a surface with a complex structure, such that complex analysis can be defined on the surface.

Suppose $f : \mathbb{C} \to \mathbb{C}$ is a complex function $f(x, y) = (u(x, y), v(x, y))$, f is *holomorphic*, if it satisfies the following *Cauchy-Riemann equations*,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}.$$

If a holomorphic function f is bijection, and the inverse f^{-1} is also holomorphic, then f is *biholomorphic*.

As shown in Fig. 4, suppose S is a surface covered by a collection of open sets $\{U_\alpha\}$, $S \subset \bigcup_\alpha U_\alpha$. A chart is (U_α, ϕ_α), where $\phi_\alpha : U_\alpha \to \mathbb{R}^2$ is a homeomorphism, and the chart transition function $\phi_{\alpha\beta} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta)$, $\phi_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1}$. The collection of the charts $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}$ is called the atlas of S. If all chart transition functions are biholomorphic, then the atlas is called a *conformal structure* of the surface.

Suppose S has a Riemannian metric $\mathbf{g}$, and all the local coordinates of the conformal structure are isothermal, then S is called a *Riemann surface*.

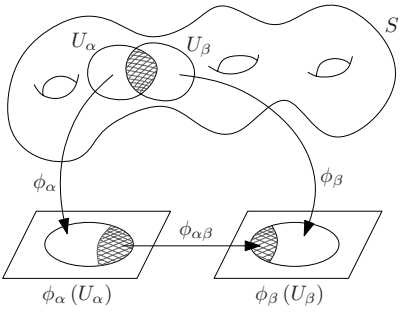


Fig. 4. A surface is covered by an atlas. If all chart transitions are holomorphic, the atlas is a conformal atlas. If all local coordinates are isothermal, the surface is a Riemann surface.

Suppose ω is a complex differential form, such that on each chart (U_α, ϕ_α) with local complex parameter z_α, $\omega = f_\alpha(z_\alpha)dz_\alpha$. Suppose two charts overlap $U_\alpha \cap U_\beta$, then

$$f_\beta(z_\beta) = f_\alpha(z_\alpha(z_\beta))\frac{dz_\alpha}{dz_\beta}.$$

Suppose $(S_1, \mathcal{A}_1)$ and $(S_2, \mathcal{A}_2)$ are two Riemann surfaces, $\mathcal{A}_i$'s are their conformal structures. Suppose (U_α, ϕ_α) is a local chart of $\mathcal{A}_1$, (V_β, ψ_β) is a local chart of $\mathcal{A}_2$. $\phi : S_1 \to S_2$ is a *conformal map* if and only if

$$\psi_\beta \circ \phi \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha) \to \psi_\beta(V_\beta)$$

is biholomorphic. A conformal map preserves angles.

4 Computational Methodology

In this section, we briefly introduce the theoretic aspects of different computational methodologies. These methods inspire the discrete computational algorithms.

4.1 Harmonic Maps

Suppose $\phi : S \to \mathbb{R}$ is a real function on the surface S, which has a metric $\mathbf{g}$, the *harmonic energy* of ϕ is defined as

$$E(\phi) = \int_S |\nabla_{\mathbf{g}}\phi|^2 dA_{\mathbf{g}}, \tag{6}$$

where $\nabla_{\mathbf{g}}$ and $dA_{\mathbf{g}}$ are the gradient operator and the area element under the metric $\mathbf{g}$, respectively. If (u_1, u_2) are isothermal coordinates with $e^{-2\lambda}$ as the conformal factor, then

$$\nabla_{\mathbf{g}} = e^{-\lambda}(\frac{\partial}{\partial u_1} \ \frac{\partial}{\partial u_2})^T, dA_{\mathbf{g}} = e^{2\lambda}du_1du_2.$$

Fig. 5. Conformal parameterization using harmonic map

A *harmonic function* is a critical point of the harmonic energy. A harmonic function satisfies the following Laplace equation,

$$\Delta_{\mathbf{g}}\phi = 0, \Delta_{\mathbf{g}} = e^{-2\lambda}(\frac{\partial^2}{\partial u_1^2} + \frac{\partial^2}{\partial u_2^2}),$$

where $\Delta_{\mathbf{g}}$ is called the *Laplace-Beltrami operator* determined by the metric $\mathbf{g}$. A harmonic map satisfies the *mean-value* property, for any point p

$$\phi(p) = \int_D \phi(q)dA,$$

where D is a small disk centered at point p.

Suppose ϕ is a map $\phi : S \to \mathbb{R}^n$, then ϕ is a harmonic map, if all its components are harmonic functions. All conformal maps must be harmonic, but the inverse is not true, except genus 0 closed surface, whose harmonic map is also conformal.

Fig. 5 illustrates a conformal parameterization result using harmonic map method.

The common way to compute a harmonic map is to construct a map first, then use the following heat flow method to diffuse the map to be harmonic:

$$\frac{d\phi}{dt} = -\Delta_{\mathbf{g}}\phi. \tag{7}$$

The heat flow is with specific constraints. For example, in order to compute the conformal map from a genus zero closed surface to the unit sphere, the nonlinear heat flow method is applied in [10].

4.2 Holomorphic 1-Forms

Definition 1 (Holomorphic 1-forms). *Suppose ω is a complex differential form, such that on each chart (U_α, ϕ_α) with local complex parameter z_α, $\omega = f_\alpha(z_\alpha)dz_\alpha$. Suppose two charts overlap $U_\alpha \cap U_\beta$, then*

$$f_\beta(z_\beta) = f_\alpha(z_\alpha(z_\beta))\frac{dz_\alpha}{dz_\beta}.$$

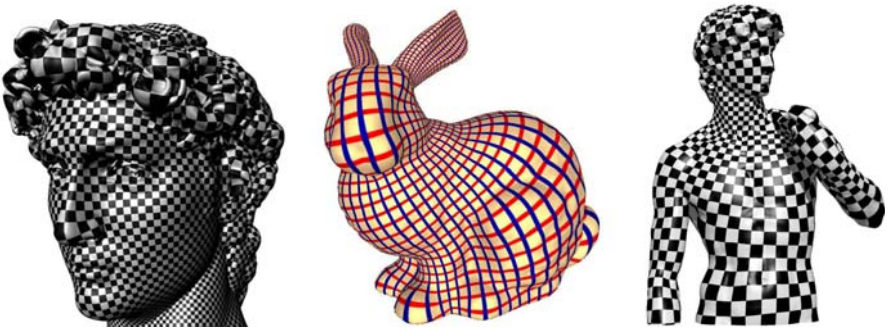


Fig. 6. Holomorphic forms on surfaces. The horizontal and vertical trajectories of the holomorphic forms are illustrated on the bunny surface. The conformal grids formed by the horizontal and vertical trajectories are demonstrated by the checker board texture mapping on the David head surface and the David body model.

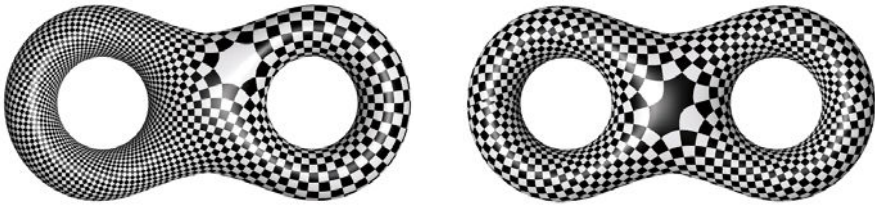


Fig. 7. The octagons in the centers are zero points of a holomorphic 1-form on a genus two surface

A holomorphic 1-form $f(z)dz$ on a Riemann surface defines a field of line elements dz, by the requirement that $f(z)dz$ is real, namely $\arg f(z) = -\arg dz \mod \pi$. The integral curves of this field of line elements are called a *horizontal trajectories*. If $f(z)dz$ is imaginary, then the integral curves of the field of line elements are called a *vertical trajectories*.

Fig. 6 illustrates the trajectories on surfaces. On the bunny surface, the horizontal and the vertical trajectories are explicitly illustrated. On the David head surface and the David body surface, the conformal grids formed by the trajectories are illustrated using checker board texture mapping.

According to the Poincaré-Hopf theorem, each holomorphic 1-form has zero points where $f(z_\alpha) = 0$. In general, the total number of zero points equals the Euler number $|\chi(S)|$. Locally, one can construct a conformal chart in a neighborhood of a zero point, such that the 1-form is $z_\alpha dz_\alpha$. Fig. 7 shows the two zero points of a holomorphic 1-form on a genus two surface.

The computational algorithm for holomorphic 1-forms is as the following: first a cohomology group basis is constructed using algebraic topological methods, then cohomology basis are diffused to be harmonic forms using the heat flow

method in Eqn. 7, finally the conjugate of harmonic forms are computed to form the holomorphic 1-forms. Details can be found in [19].

4.3 Ricci Flow

Suppose S is a smooth surface with a Riemannian metric $\mathbf{g}$. The Ricci flow is the process to deform the metric $\mathbf{g}(t)$ by its induced Gaussian curvature $K(t)$, where t is the time parameter, such that the curvature evolves according to a heat diffusion,

$$\frac{dg_{ij}(t)}{dt} = -2K(t)g_{ij}(t). \tag{8}$$

Suppose $T(u_1, u_2, t)$ is a temperature field on the surface, then according to the thermal dynamics, the temperature field will evolve governed by the following heat diffusion equation,

$$\frac{T(t)}{dt} = -\Delta_{\mathbf{g}} T(t).$$

In Ricci flow, the curvature evolution is exactly the same as heat diffusion on the surface, as follows:

$$\frac{K(t)}{dt} = -\Delta_{\mathbf{g}(t)} K(t), \tag{9}$$

where $\Delta_{\mathbf{g}(t)}$ is the Laplace-Beltrami operator induced by the metric $\mathbf{g}(t)$. We can simplify the Ricci flow equation 8. Let $g(t) = e^{2\lambda(t)} g(0)$, then Ricci flow is

$$\frac{d\lambda(t)}{dt} = -2K(t). \tag{10}$$

The following theorems postulate that the Ricci flow defined in Eqn. 8 is convergent and lead to the conformal uniformization metric.

Theorem 2 (Hamilton 1982 [34]). *For a closed surface of non-positive Euler characteristic, if the total area of the surface is preserved during the flow, the Ricci flow will converge to a metric such that the Gaussian curvature is constant everywhere.*

Theorem 3 (Chow 1991 [39]). *For a closed surface of positive Euler characteristic, if the total area of the surface is preserved during the flow, the Ricci flow will converge to a metric such that the Gaussian curvature is constant everywhere.*

The corresponding metric $\mathbf{g}(\infty)$ is the *uniformization metric*. Moreover, at any time t, the metric $\mathbf{g}(t)$ is conformal to the original metric $\mathbf{g}(0)$.

The Ricci flow can be easily modified to compute a metric with a *prescribed* curvature $\bar{K}$, and then the flow becomes

$$\frac{dg_{ij}(t)}{dt} = 2(\bar{K} - K)g_{ij}(t). \tag{11}$$

With this modification, any target curvatures $\bar{\mathbf{K}}$, which are admissible with the Gauss-Bonnet theorem, can be induced from the solution metric $\mathbf{g}(\infty)$.

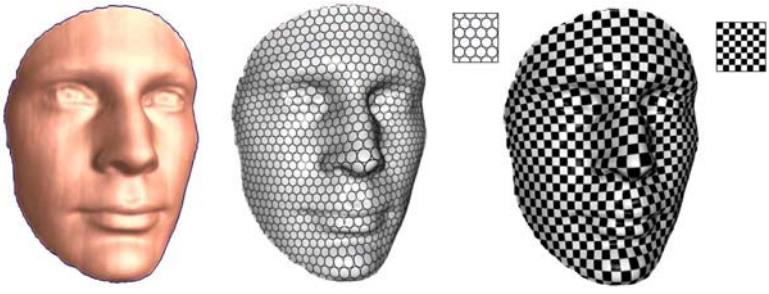


Fig. 8. Conformal parameterizations map infinitesimal circles to infinitesimal circles and preserve the intersection angles between the circles

5 Discrete Ricci Flow

In engineering fields, smooth surfaces are often approximated by simplicial complexes (triangle meshes). We consider a triangle mesh Σ with vertex set V, edge set E and face set F.

We say Σ is with Euclidean background geometry, or simply Σ is an *Euclidean mesh*, if each of its faces is realizable on the Euclidean plane. Similarly, we can define meshes with spherical or hyperbolic background geometries respectively.

The relations among angles and edge lengths are governed by cosine laws. Meshes with different background geometries require the corresponding cosine laws.

Discrete Riemannian Metric. In the discrete setting, the edge lengths on a mesh Σ simply define the *discrete Riemannian metric* on Σ,

$$l : E \to \mathbb{R}^+,$$

such that for a face f_{ijk} the edge lengths satisfy the triangle inequality,

$$l_{ij} + l_{jk} > l_{ki}.$$

The discrete metric determines the angles. Suppose we have a triangle f_{ijk} with edge lengths $\{l_{ij}, l_{jk}, l_{ki}\}$, and the angles against the corresponding edges are $\{\theta_k, \theta_i, \theta_j\}$. The cosine laws with respect to the background geometries are

$$\begin{array}{ll} l_{ij}^2 & = l_{jk}^2 + l_{ki}^2 - 2l_{jk}l_{ki}\cos\theta_k, \quad \mathbb{E}^2 \\ \cosh l_{ij} & = \cosh l_{jk}\cosh l_{ki} - \sinh l_{jk}\sinh l_{ki}\cos\theta_k, \quad \mathbb{H}^2 \\ \cos l_{ij} & = \cos l_{jk}\cos l_{ki} + \sin l_{jk}\sin l_{ki}\cos\theta_k. \quad \mathbb{S}^2 \end{array} \tag{12}$$

Discrete Gaussian Curvature. The discrete Gauss curvature is defined as the angle deficit on a mesh,

$$K_i = \begin{cases} 2\pi - \sum_{f_{ijk}\in F} \theta_i^{jk}, \text{ interior vertex} \\ \pi - \sum_{f_{ijk}\in F} \theta_i^{jk}, \text{ boundary vertex} \end{cases} \tag{13}$$

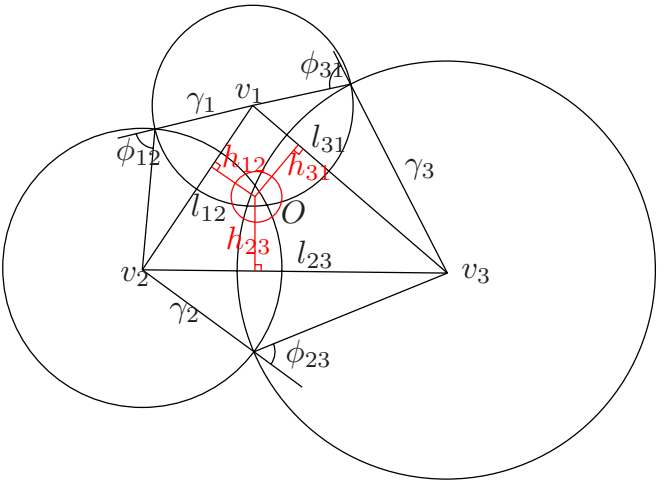


Fig. 9. Circle Packing Metric: Circle packing metric for a triangle. The center circle is the dual circle, which is orthogonal to all the other three circles.

where θ_i^{jk} represents the corner angle attached to vertex v_i in the face f_{ijk}.

Discrete Gauss-Bonnet Theorem. In the discrete setting, the Gauss-Bonnet theorem (Eqn. 4) still holds on meshes with the discrete Gaussian curvatures, as follows.

$$\sum_{v_i \in V} K_i + \lambda \sum_{f_i \in F} A_i = 2\pi\chi(M),$$

where A_i denotes the area of face f_i, and λ represents the constant curvature for the canonical geometry: $+1$ for sphere, 0 for plane, and -1 for hyperbolic space.

Discrete Conformal Deformation

In order to approximate conformal deformation of metrics, the circle packing metric is introduced [27,40]. Let us denote Γ as a function which assigns a radius γ_i to each vertex v_i.

$$\Gamma : V \to \mathbb{R}^+$$

Similarly, let a *weight* on the mesh be a function,

$$\Phi : E \to [0, \frac{\pi}{2}].$$

on each edge e_{ij} by assigning a positive number $\Phi(e_{ij})$. The pair of vertex radius and edge weight functions on a mesh Σ, (Γ, Φ), is called a *circle packing metric* of Σ. Figure 9 illustrates the circle packing metrics. Each vertex v_i has a circle whose radius is r_i. On each edge e_{ij}, an intersection angle ϕ_{ij} is defined by two circles of v_i and v_j, which are intersected with or tangent to each other. Two circle packing metrics (Γ_1, Φ_1) and (Γ_2, Φ_2) on a same mesh are *conformal equivalent*, if $\Phi_1 \equiv \Phi_2$. Therefore, a conformal deformation of a circle packing metric only modifies the vertex radii.

For a given mesh, its circle packing metric and the edge lengths on the mesh can be converted to each other by using cosine laws. Here, the choice of the cosine laws is dependent on the background geometry of the mesh,

$$\begin{array}{rl} l_{ij}^2 = \gamma_i^2 + \gamma_j^2 + 2\gamma_i\gamma_j \cos\phi_{ij}, & \mathbb{E}^2 \\ \cosh l_{ij} = \cosh\gamma_i \cosh\gamma_j + \sinh\gamma_i \sinh\gamma_j \cos\phi_{ij}, & \mathbb{H}^2 \\ \cos l_{ij} = \cos\gamma_i \cos\gamma_j - \sin\gamma_i \sin\gamma_j \cos\phi_{ij}. & \mathbb{S}^2 \end{array} \tag{14}$$

5.1 Theories on Discrete Ricci Flow

The connections between the circle packing metric and the discrete Gaussian curvature can be obtained from Eqn. 14 and 13. Now, we are ready to explain the discrete Ricci flow.

Let u_i to be

$$u_i = \begin{cases} \log\gamma_i & \mathbb{E}^2 \\ \log\tanh\frac{\gamma_i}{2} & \mathbb{H}^2 \\ \log\tan\frac{\gamma_i}{2} & \mathbb{S}^2 \end{cases} \tag{15}$$

for each vertex. Then, the discrete Ricci flow is defined as follows,

$$\frac{du_i(t)}{dt} = (\bar{K}_i - K_i). \tag{16}$$

Recall that, in continuous cases, Riemannian metrics determine the Gaussian curvature, and continuous Ricci flow is the conformal deformation of the Riemannian metric such that the deformation is proportional to the Gaussian curvature. Similarly, in discrete case, the circle packing metric determines the discrete Gaussian curvature (Eqn. 14, 13, and 12), and the discrete Ricci flow conformally deforms the circle packing metrics with respect to the Gaussian curvatures (Eqn. 15 and 16).

Discrete Ricci flow can be formulated in the variational setting, namely, it is a negative gradient flow of some special energy form. Let Σ be a spherical (Euclidean or hyperbolic) triangle mesh, then for arbitrary two vertices v_i, v_j, the following symmetric relation holds

$$\frac{\partial K_i}{\partial u_j} = \frac{\partial K_j}{\partial u_i}.$$

Let $\omega = \sum_{i=1}^n K_i du_i$ be a differential one-form [41]. The symmetric relation guarantees the one-form is closed (curl free),

$$d\omega = \sum_{i,j}(\frac{\partial K_i}{\partial u_j} - \frac{\partial K_j}{\partial u_i})du_i \wedge du_j = 0.$$

Furthermore, the vertex radii $\mathbf{u}$ domain is simply connected. By Stokes theorem, the following integration is path independent,

$$f(\mathbf{u}) = \int_{\mathbf{u_0}}^{\mathbf{u}} \sum_{i=1}^n (\bar{K}_i - K_i)du_i, \tag{17}$$

where $\mathbf{u_0}$ is an arbitrary initial metric. Therefore, the above integration is well defined, and called the *Ricci energy*. The discrete Ricci flow is the negative gradient flow of the discrete Ricci energy. The discrete metric which induces $\bar{k}$ is the minimizer of the energy.

Computing desired metric with prescribed curvature $\bar{K}$ is equivalent to minimizing the discrete Ricci energy. For Euclidean (or hyperbolic) case, the discrete Ricci energy is strictly convex (namely, its Hessian is positive definite). The global minimum uniquely exists, corresponding to the metric $\bar{\mathbf{u}}$, which induces $\bar{\mathbf{k}}$. The discrete Ricci flow converges to this global minimum [11]. The Euclidean Ricci energy is strictly convex on the space of normalized metric $\sum u_i = 0$. The hyperbolic Ricci energy is strictly convex. The Spherical Ricci energy is not strictly convex, the desired metric $\bar{\mathbf{u}}$ is still a critical point of the energy and can be reached by the Ricci flow (gradient descent) method.

6 Applications

Discrete Ricci flow is a powerful tool for computing the desired metrics with the prescribed Gaussian curvatures on general surfaces. Many applications in graphics and geometric modeling can be formulated as the problem of finding specific metrics for prescribed curvatures. The followings are some applications directly related with discrete Ricci flow.

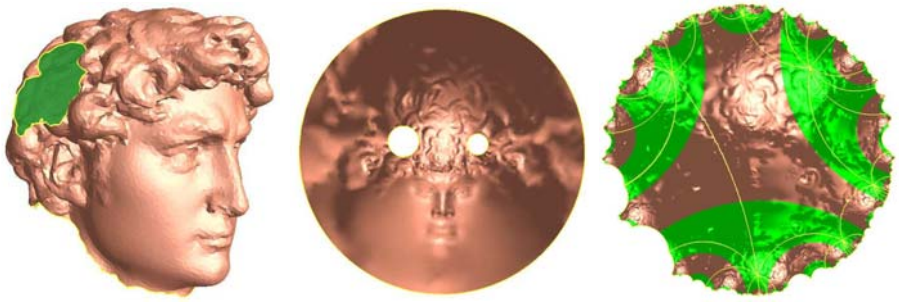
6.1 Global Conformal Parameterization

Surface parameterization in conventional computer graphics refers to the process of mapping a surface to a canonical domain. Ricci flow is an effective approach for fully automatic, seam-free, and singularity-free parameterization of arbitrarily complicated surfaces.

For closed surfaces, we compute their uniformization metric and embed their universal covering spaces onto the three canonical spaces, as described in the uniformization theorem. Fig. 1 illustrates the global parameterizations of surfaces with different topologies with their uniformization metrics respectively.

For surfaces with boundaries, there are two general ways. In the first way, we map the interior of the surface to planar domains, and the boundaries to circles. In the second way, the surface is embedded onto the canonical spaces with its uniformization metric, such that all the boundaries are mapped to geodesics. Fig. 10 shows one example, where a genus zero surface with three boundaries are parameterized in both ways.

In the general case, both the positions and curvatures of singularities can be arbitrarily chosen, as long as the Gauss-Bonnet condition and the connectivity constraints hold, the metric can be obtained using Ricci flow method. The flexibility of assigning singularities greatly improves the quality of conformal parameterizations. Fig. 11 demonstrates the parameterizations of scanned surfaces using this method.



(a) (b) (c)

Fig. 10. (a) The David head surface is a two-holed disk. (b) It is parameterized as a planar multi-holed annulus. (c) It can also be parameterized in the Poincaré disk, such that all boundaries are mapped to geodesics.

(a) (b) (c)

Fig. 11. Parameterizations of scanned surfaces using discrete Ricci flow

6.2 Surface Matching

Surface matching is a fundamental problem in geometric processing. The main framework of surface matching using Ricci flow can be formulated in the following commutative diagram,

$$\begin{array}{ccc} S_1 & \xrightarrow{\phi} & S_2 \\ {\scriptstyle \tau_1}\downarrow & & \downarrow{\scriptstyle \tau_2} \\ D_1 & \xrightarrow{\bar{\phi}} & D_2 \end{array} \tag{18}$$

S_1, S_2 are two given surfaces, $\phi : S_1 \to S_2$ is the desired matching. We use Ricci flow to compute $\tau_i : S_i \to D_i$ which maps S_i conformally onto the canonical domain D_i. D_1 and D_2 can also be surfaces other than simple planar domains. The topology and the curvature of D_1 and D_2 incorporate the major feature information of the original surfaces S_1 and S_2. Then we construct a harmonic map

Fig. 12. Two genus one surfaces with 3 bounaries are conformally parameterized onto the plane, such that their boundaries are mapped to circles. Then a map is constructed between their planar images, which induces a matching between them. The matching is visualized by the consistent texture mapping.

$\bar{\phi} : D_1 \to D_2$. If there are certain feature constraints, they can be incorporated into $\bar{\phi}$. The final map ϕ is induced by $\phi = \tau_2^{-1} \circ \bar{\phi} \circ \tau_1$. Fig. 12 shows one example of surface matching between two genus one surfaces with three boundaries. Both of them are mapped onto the plane with circular boundaries, D_1 and D_2 chosen to be their fundamental domains. The harmonic map is calculated between the two fundamental domains. The mapping result is illustrated by consistent texture mapping.

6.3 Computing General Geometric Structures

Suppose S is a surface, $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}$ is an atlas of S. If all the local coordinates $\phi_\alpha(U_\alpha)$ is in a special space X, and all the chart transition functions $\phi_{\alpha\beta}$ to a special group G, which is a subgroup of the automorphism group of X, then $\mathcal{A}$ is called a (X, G) structure of S.

All genus zero closed surfaces can be conformally mapped onto the sphere, namely, they admit spherical structures (see Fig. 1).

Suppose X is the two dimensional affine space $\mathbb{A}^2$, and G is the affine transformation group, then the atlas is called an *Affine Structure* of S. Affine structure plays fundamental role in manifold splines. For closed surfaces, only genus one surfaces have affine structures. All surfaces with boundaries admit affine structures.

If X is the hyperbolic space $\mathbb{H}^2$, and G is the hyperbolic rigid motion (Möbius transformation), then the atlas is called a *hyperbolic structure*. Surfaces with negative Euler number admit hyperbolic structures. Fig. 13 demonstrates the hyperbolic structures of genus two surfaces.

All surfaces have real projective structures, where the coordinates are in the real projective space $\mathbb{RP}^2$, and the chart transition functions are real projective transformations. For high genus surfaces, the real projective structure can be constructed by using hyperbolic Ricci flow to compute the hyperbolic uniformization metric, then isometrically embedding the universal covering space onto the Klein hyperbolic space model.

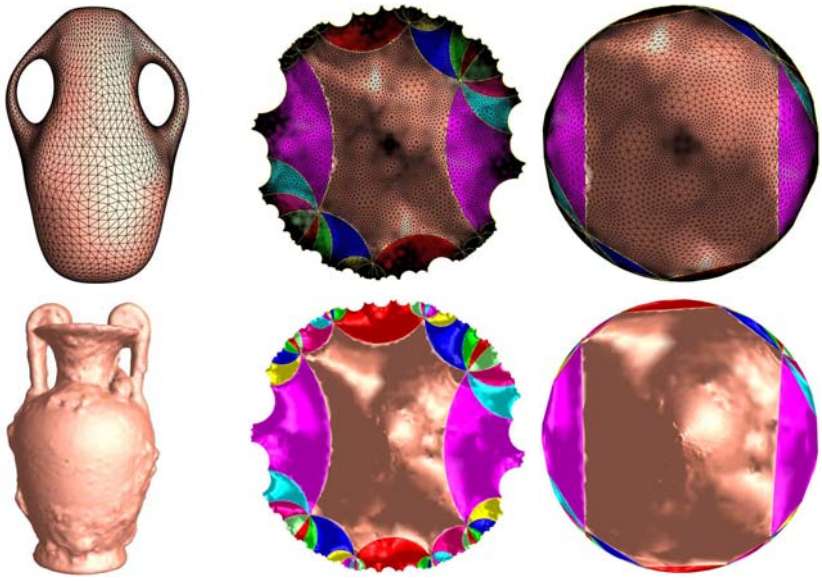


Fig. 13. Hyperbolic structure and real projective structure of genus two surfaces

6.4 Manifold Splines

Conventional splines are defined on planar domains. However, it is natural to define splines directly on surfaces with general topologies. The concept of manifold splines was first introduced in [9], where the splines are defined on manifold domains and the evaluations of the splines are independent of the choice of the local charts.

The significant advantage of the manifold spline is that it is globally defined, but locally on each chart, a common planar spline. As it has been proved that defining splines over arbitrary manifolds is equivalent to the existence of an affine atlas of the underlying manifold. The affine atlas can be constructed by holomorphic 1-form method [19], as described in [9]. The positions of zero points can not be specified, which are solely determined by the conformal structure and the holomorphic 1-form. The affine structure can also be constructed using Ricci flow, while the singularities can be arbitrarily chosen.

Fig. 14 shows this problem with genus three sculpture model, which has to carry four zero points when constructing manifold spline (see more details in [9]). Even for genus-1 surfaces with boundaries, since differential forms method does not work directly on open surfaces, the construction of manifold splines is still not able to avoid singular points after double covering which converts open surfaces to closed ones.

With Euclidean Ricci flow, all the curvatures of a given domain manifold can be put on one vertex or boundaries, such that the special metric of the domain manifold is flat everywhere except at one singular point. Then, the metric

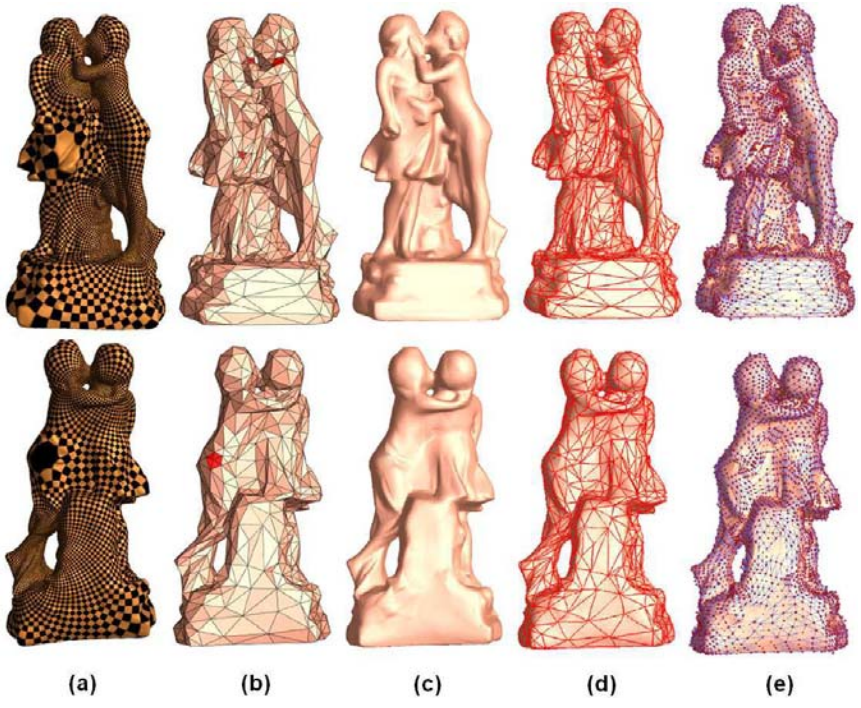


Fig. 14. Manifold Spline of Sculpture Model Constructed Using Differential Forms With 4 Singular Points: (a) Holomorphic 1-form which induces the affine atlas $\mathbb{A}$; (b) Parametric domain manifold M with singular points marked; (c) Polynomial spline F defined on the manifold M in (a); (d) The red curves on spline F correspond to the edges in the domain manifold M; (e) Spline F covered by control net C

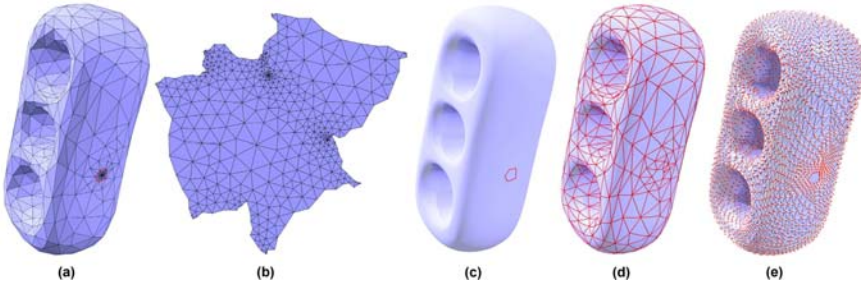


Fig. 15. Manifold Spline of Genus-3 Model Constructed Using discrete Euclidean Ricci Flow With Only One Singular Point: (a) Domain manifold (b) Central chart (c) Manifold triangular B-spline (d) Manifold triangular B-spline (e) Control points

induces an affine atlas covering the whole manifold except the singular point, and manifold spline is constructed over this affine atlas. Fig. 15 shows the an example of a genus-3 model with only one singular point when constructing manifold spline using Euclidean Ricci flow (see [36] for further details).

7 Conclusion and Future Works

This work introduces a powerful tool: discrete surface Ricci flow, borrowed from continuous Ricci flow from modern geometry for computing Riemannian metrics with prescribed Gaussian curvatures on discrete surfaces. For practical use, the concepts of Riemannian metric and Gaussian curvature are translated to discrete setting; the discrete analogy of Ricci flow is designed; the existence and uniqueness of the solution of the discrete Ricci flow is verified; and efficient numerical algorithms to compute Riemannian metrics with prescribed Gaussian curvatures using discrete Ricci flow for generally topological surfaces are presented in details.

Several applications in graphics and geometric modeling, such as surface parameterization, surface matching, manifold splines, and construction of geometric structures on general surfaces are demonstrated.

The discrete theory and computational methodologies for Teichmüller space theory are widely open. The rigorous theoretic results about the convergence to the smooth conformal structure and the approximation accuracy of discrete Ricci flow have not been established. In the future, we will continue our exploration along these directions.

Acknowledgements

The authors would like to thank Prof. Shing-Tung Yau, Prof. Tom Sederberg, Prof. Ralph Martin and Prof. Hong Qin for discussions. This work was supported by NSF 0448399 career: Conformal Geometry Applied to Shape Analysis and Geometric Modeling, NSF 0528363: conformal geometry in graphics and visualization, and NSF 0626223: discrete curvature flow.

References

1. Perelman, G.: The entropy formula for the ricci flow and its geometric applications. Technical Report arXiv.org (2002)
2. Perelman, G.: Ricci flow with surgery on three-manifolds. Technical Report arXiv.org (2003)
3. Perelman, G.: Finite extinction time for the solutions to the ricci flow on certain three-manifolds. Technical Report arXiv.org (2003)
4. Hamilton, R.S.: Three manifolds with positive ricci curvature. Journal of Differential Geometry. 17, 255–306 (1982)
5. Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. SIGGRAPH 2002, pp. 362–371 (2002)

6. Botsch, M., Kobbelt, L.: An interactive framework for real-time freeform modeling. ACM Transactions on Graphics (SIGGRAPH 2004) 23(3), 630–634 (2004)
7. Alliez, P., Meyer, M., Desbrun, M.: Interactive geometry remeshing. SIGGRAPH 2002, pp. 347–354 (2002)
8. Grimm, C., Zorin, D.: Surface Modeling and Parameterization with Manifolds. SIGGRAPH 2006 course notes (2006)
9. Gu, X., He, Y., Qin, H.: Manifold splines. Graphical Models 68(3), 237–254 (2006)
10. Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.T.: Genus zero surface conformal mapping and its application to brain surface mapping. IEEE Transactions on Medical Imaging 23(8), 949–958 (2004)
11. Chow, B., Luo, F.: Combinatorial ricci flows on surfaces. Journal Differential Geometry 63(1), 97–129 (2003)
12. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: Advances in Multiresolution for Geometric Modelling, pp. 157–186. Springer, Heidelberg (2005)
13. Kraevoy, V., Sheffer, A.: Cross-parameterization and compatible remeshing of 3d models. ACM Transactions on Graphics 23(3), 861–869 (2004)
14. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. Computer Graphics Forum (Proc. Eurographics 2002) 21(3), 209–218 (2002)
15. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental Mathematics 2(1), 15–36 (1993)
16. Floater, M.S.: Mean value coordinates. Computer Aided Geometric Design 20(1), 19–27 (2003)
17. Zayer, R., Rössl, C., Seidel, H.P.: Setting the boundary free: A composite approach to surface parameterization. In: Pottmann, M.D., H. (eds.) Eurgraphics Symposium on Geometry Processing, pp. 91–100 (2005)
18. Gotsman, C., Gu, X., Sheffer, A.: Fundamentals of spherical parameterization for 3d meshes. ACM Transactions on Graphics 22(3), 358–363 (2003)
19. Gu, X., Yau, S.T.: Global conformal parameterization. In: Symposium on Geometry Processing, pp. 127–137 (2003)
20. Mercat, C.: Discrete riemann surfaces and the ising model. Communications in Mathematical Physics 218(1), 177–216 (2004)
21. Hirani, A.N.: Discrete exterior calculus. PhD thesis, California Institute of Technology (2003)
22. Jin, M., Wang, Y., Yau, S.T., Gu, X.: Optimal global conformal surface parameterization. In: IEEE Visualization 2004, pp. 267–274 (2004)
23. Gortler, S.J., Gotsman, C., Thurston, D.: Discrete one-forms on meshes and applications to 3D mesh parameterization. Computer Aided Geometric Design 23(2), 83–112 (2005)
24. Tong, Y., Alliez, P., Cohen-Steiner, D., Desbrun, M.: Designing quadrangulations with discrete harmonic forms. In: Symposium on Geometry Processing, pp. 201–210 (2006)
25. Sheffer, A., de Sturler, E.: Parameterization of faced surfaces for meshing using angle based flattening. Engineering with Computers 17(3), 326–337 (2001)
26. Sheffer, A., Lévy, B., Mogilnitsky, M., Bogomyakov, A.: ABF++: Fast and robust angle based flattening. ACM Transactions on Graphics 24(2), 311–330 (2005)
27. Thurston, W.P.: Geometry and Topology of Three-Manifolds. Princeton lecture notes (1976)
28. Rodin, B., Sullivan, D.: The convergence of circle packings to the riemann mapping. Journal Differential Geometry 26(2), 349–360 (1987)

29. Collins, C., Stephenson, K.: A circle packing algorithm. Computational Geometry: Theory and Applications 25, 233–256 (2003)
30. de Verdiére, Y.C.: Un principe variationnel pour les empilements de cercles. Journal Differential Geometry 104(3), 655–669 (1991)
31. Bobenko, A.I., Springborn, B.A.: Variational principles for circle patterns and koebe's theorem. Transactions of the American Mathematical Society 356, 659–689 (2004)
32. Kharevych, L., Springborn, B., Schröder, P.: Discrete conformal mappings via circle patterns. ACM Transactions on Graphics 25(2), 412–438 (2006)
33. Springborn, B.: Variational Principles for Circle Patterns. PhD thesis, Technische Universität Berlin (2003)
34. Hamilton, R.S.: The ricci flow on surfaces. Mathematics and general relativity 71, 237–262 (1988)
35. Jin, M., Luo, F., Gu, X.: Computing surface hyperbolic structure and real projective structure. In: ACM Symposium on Solid and Physics Modeling, pp. 105–116 (2006)
36. Gu, X., He, Y., Jin, M., Luo, F., Qin, H., Yau, S.T.: Manifold splines with single extraordinary point. In: ACM Symposium on Solid and Physics Modeling (to appear 2007)
37. Guggenheimer, H.W.: Differential Geometry. Dover Publications, Mineola (1977)
38. Farkas, H.M., Kra, I.: Riemann Surfaces. Springer, Heidelberg (2004)
39. Chow, B.: The ricci flow on the 2-sphere. J. Differential Geom. 33(2), 325–334 (1991)
40. Stephenson, K.: Introduction To Circle Packing. Cambridge University Press, Cambridge (2005)
41. Weitraub, S.H.: Differential Forms: A Complement to Vector Calculus. Academic Press, San Diego (2007)

Guided C^2 Spline Surfaces with V-Shaped Tessellation

Kestutis Karčiauskas[1] and Jorg Peters[2]

[1] Dept. of Mathematics and Informatics, Vilnius University, Lithuania
kestutis.karciauskas@maf.vu.lt
[2] Dept. CISE, University of Florida, USA
jorg@cise.ufl.edu

Abstract. The guided spline approach to surface construction separates surface design and surface representation by constructing local guide surfaces and sampling these by splines of moderate degree. This paper explains a construction based on tessellating the domain into V-shaped regions so that the resulting C^2 surfaces have G^2 transitions across the boundaries of the V-shapes and consist of tensor-product splines of degree (6,6) with patches of degree (4,4) forming a central cap.

1 Introduction

Our goal is to smoothly fill or cap a multi-sided hole in a C^2 spline complex by a (small) finite number of spline patches and without introducing shape artifacts such as curvature fluctuation or loss of convexity. The surface cap will interpolate position, first and second derivative for spline boundary data up to degree 6. This degree permits exact blends not only in the common setting of bicubic patch complexes, but also when joining features like cylinders and tori (see Figure 12). We follow the general strategy laid out and illustrated in [1]: we separate surface design and surface representation by constructing local guide surfaces and sampling these by splines of moderate degree. For a guide surface $\mathbf{g} : \mathbb{R}^2 \to \mathbb{R}^3$, a tessellation map $\tilde{\rho} : \mathbb{R}^2 \to \mathbb{R}^2$, the sampling operator $H : \mathbf{g} \circ \tilde{\rho} \mapsto \mathbf{x}$ maps the composition of the guide and the tessellation map to the final surface representation $\mathbf{x} := H(\mathbf{g} \circ \tilde{\rho})$. We can interpret this as H deciding the formal continuity and $\mathbf{g}$ expressing the design intent. In particular, $\mathbf{g}$ is a surface piece that outlines the intended shape, but need not fit the requirements of a typical modeling pipeline due to its representation or lack of continuity with the existing data. H takes data from $\mathbf{g}$ to approximately rebuild $\mathbf{g}$ but in a framework suitable for standard design cycles. The basic construction options for H and $\tilde{\rho}$ have been laid out in detail in [1,2]. The remaining, nontrivial technical challenge in devising surface caps is to find good transitions $\tau : \mathbb{R}^2 \to \mathbb{R}^2$ between maps ρ and σ, where $\rho : \mathbb{R}^2 \to \mathbb{R}^2$ is the parameterization of the surrounding data and $\sigma : \mathbb{R}^2 \to \mathbb{R}^2$ tessellates the immediate neighborhood of the pre-image of a central point (see Figure 2, dark, blue areas).

Two finite constructions of this type have been devised in [1,3]. In [1], (Figure 2, *left*), σ and τ are of degree (5,5) and the key to the construction

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 233–244, 2007.
© Springer-Verlag Berlin Heidelberg 2007

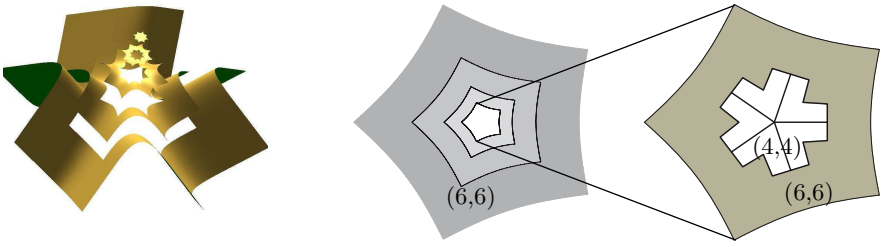


Fig. 1. (*left*) Sequence of guided surface rings and cap in $\mathbb{R}^3$, offset to show the structure. (*middle*) Surface rings of degree (6,6) constructed according to [2]. (*right*) The focus of this paper: transition ring of degree (6,6) and final cap consisting of n V-shaped surface pieces of degree (4,4).

is that the partial derivatives $\partial^k\sigma$, $k = 0, 1, 2$, of σ on the n sector partitioning edges (cf. Figure 3, *left*) of the n-sided cap are of degree $2k + 1$. There are $3n$ polynomial pieces in τ. In [3], (Figure 2, *middle*), σ and τ are of degree (3,3) and (4,4) respectively, and there are $30n$ polynomial pieces in τ. The key technical achievement of this complex construction is that the partial derivatives $\partial^k\sigma$ across the outer n-gon boundaries of σ are of degree $k + 1$. This yields patches of degree (4,4) surrounding the central point. The present paper introduces a third parameterization, (Figure 2, *right*), closely related to second construction, but simpler and unusual in its layout (see Figure 1). It results in $16n$ polynomial pieces in τ, such that the outer boundaries of σ are of degree $k+1$. Again, this yields C^2 surfaces with splines of degree (6,6) in the transition and patches of degree (4,4) surrounding the central point. The unusual V-shaped layout is inherited from the tessellation of the neighborhood of the pre-image of the central point. This tessellation is described in Section 2. The tessellation needs to transition smoothly to a standard parameterization of surrounding data for the transition between central cap and guided approach to work. The transition tessellation is defined in Section 3. Section 4 then defines the surface cap. The advantage of the present construction

- compared to [1] is the lower overall degree,
- compared to [3] the halving of the number of patches, simpler layout and stability for higher valences.

2 A V-Shaped Tessellation Map

The basic idea underlying the tessellation was introduced in [4] in order to generate C^2 surfaces of least polynomial degree. While the approach advanced our technical repertoire, the accompanying algorithm was of limited use due to bad geometric properties for valences above 6. Specifically, the low degree was based on a parameterization of the unit n-gon and, for each corner, the independent

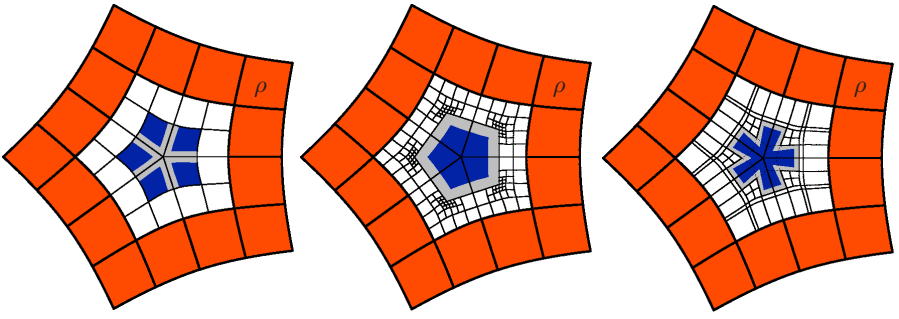


Fig. 2. Three different tessellations used respectively for surface constructions of degree $(8,8)$ [1], $(6,6)$ [3] and $(6,6)$ [this paper]. The $3n$ quadrilaterals (quads) of the outermost (*orange*) rings are parametrized by maps ρ of degree (3,3), for example the characteristic map of Catmull-Clark subdivision. The innermost (*blue*) quads are parametrized by a map σ, or, alternatively, (*middle* and *right*) by a bilinear map β. The transition regions are parametrized by maps τ. The edges with grey underlay indicate the technically most interesting transitions, the G^2 transitions. (*left*) Tessellation according to [1]: $\deg(\sigma) = \deg(\tau) = (5,5)$. (*middle*) Tessellation according to [3]: $\deg(\sigma) = (3,3)$, $\deg(\beta) = (1,1)$ and $\deg(\tau) = (4,4)$. (*right*) Tessellation according to this paper: $\deg(\sigma) = (3,3)$, $\deg(\beta) = (1,1)$ and $\deg(\tau) = (4,4)$.

local parameters u and v were associated with the directions of consecutive n-gon edges. Unfortunately, the angle between consecutive edges approaches $180°$ as the valence increases, leading in the limit for $n \to \infty$ to singularity, and much earlier to self-intersections of the Bézier control structure associated with the prolongation τ_e of the parameterization across the vertices (see Figure 3, *middle top*). [3] proposed a solution to this degeneracy, based on the observation that a finer partition allows to locally resolve the near-singularity. However, the construction is complex and applies reasonably only up to $n = 8$ (which, in practice, usually suffices).

Here, we propose a more efficient solution of this geometric problem keeping only the algebraic flavor of [4]. The idea is to cut out the obtuse angles and arrive at a V-shaped domain for each sector of σ as in Figure 3 *right*. We parametrize along $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_c, \bar{\mathbf{v}}_1, \bar{\mathbf{v}}_0$ by σ so that the boundary is piecewise linear, the first transversal derivative piecewise quadratic and only the second transversal derivative is of actual degree 3. Along the edge $\mathbf{v}_0, \mathbf{v}_1$ (Figure 4 *left*), the map consists of a single piece, along $\mathbf{v}_1, \mathbf{v}_c$ of two pieces to be able to create a consistent match at $\mathbf{v}_c$. The pieces are C^2 connected and, in particular, the pieces join correctly and are consistent at $\mathbf{v}_c$ (Figure 4 *middle*) and at the corners $\mathbf{v}_1$ (Figure 4 *right*).

The parameterization is chosen symmetric with respect to rotation by one sector. Each sector map σ is symmetric with respect to the bisectrix of the sector. We abbreviate

$$\alpha := \frac{2\pi}{n},\ \mathsf{c} := \cos\alpha,\ \mathsf{s} := \sin\alpha,\ \mathbf{0} = (0,0),\ \hat{\mathsf{c}} := \cos\frac{\alpha}{2},\ \hat{\mathsf{s}} := \sin\frac{\alpha}{2}\ .$$

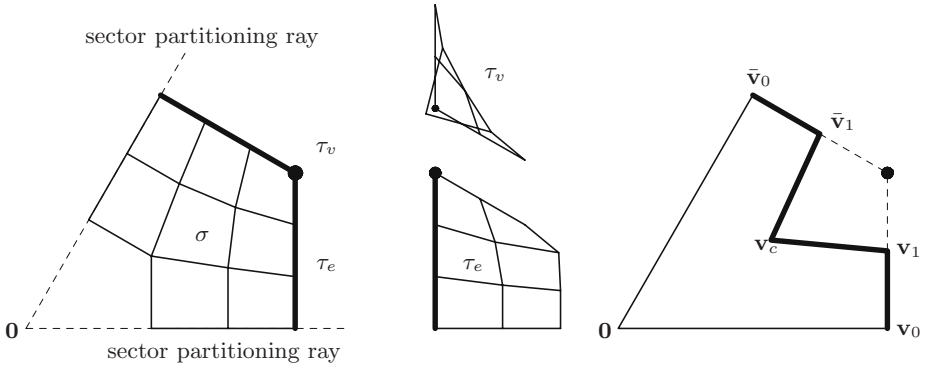


Fig. 3. One sector, valence $n = 6$: (*left*) The fat edges of σ form a sector of a n-gon. (*middle*) The prolongations of σ according to [4] (*top*) τ_v across the corner opposite the origin $\mathbf{0}$ and (*bottom*) τ_e across an edge. (*right*) New V-shaped domain after cutting out the obtuse angle of the n-gon.

A sector is bounded by the *sector partitioning rays* $\overline{\mathbf{0},(1,0)}$ and $\overline{\mathbf{0},(\mathsf{c},\mathsf{s})}$ and the V-shaped domain is the polygon (cf. Figure 3 *right*)

$$(\mathbf{0},\mathbf{v}_0,\mathbf{v}_1,\mathbf{v}_c,\bar{\mathbf{v}}_1,\bar{\mathbf{v}}_0), \qquad \mathbf{v}_0 := (z_0,0),\ \mathbf{v}_1 := (z_0,z_1),\ \mathbf{v}_c := (z_2\hat{\mathsf{c}},z_2\hat{\mathsf{s}})\ ,$$

i.e. in terms of three parameters z_0, z_1, z_2. We split the V-shape into three quadrilaterals as shown in Figure 6, *left*.

Later on, we parameterize the V-shape by three bilinear maps β, defined by the boundaries of the quadrilaterals. Here we focus on a higher-order parame-

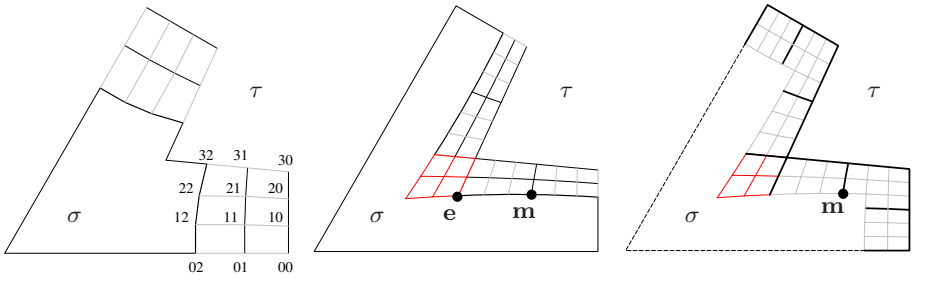


Fig. 4. (*left*) Coefficients of σ at the tip of the V-shape. (*middle*) Two polynomial pieces of σ along the edge $\mathbf{v}_1\ \mathbf{v}_c$ and their symmetric counterparts along $\bar{\mathbf{v}}_1\ \mathbf{v}_c$ join to form consistent C^2 data at $\mathbf{v}_c$. (*right*) An extra subdivision at the V-tips illustrates the uniform 'tensor-border' strip of the second derivatives prescribed by the tessellation σ.

terization σ that plays a crucial role in the construction. We need only construct the derivatives of σ up to second order, across the border $(\mathbf{v}_0,\mathbf{v}_1,\mathbf{v}_c,\bar{\mathbf{v}}_1,\bar{\mathbf{v}}_0)$ delineated as fat lines in Figure 3, *right*. These 'tensor-border' data will define

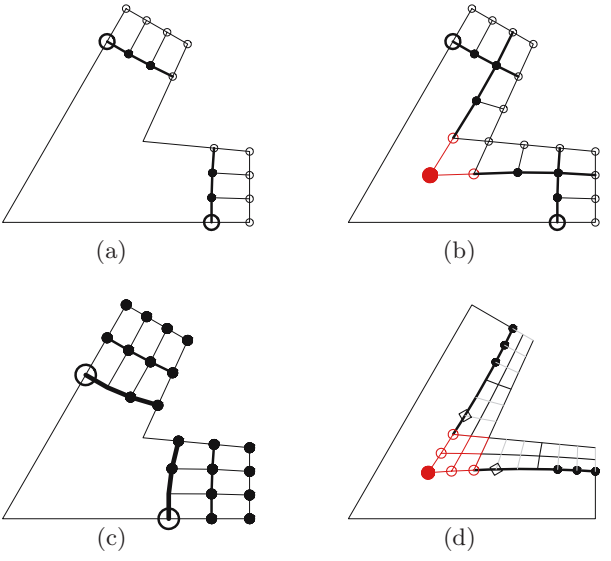


Fig. 5. Construction of σ. (a) and (b): quadratic first derivative layer: the two quadratic pieces are one subdivided quadratic. (a) The large circled point on sector partitioning ray is free. Small circles define linear maps in degree-raised form. (b) The Bézier coefficients of the C^1 extensions of the quadratic layers along $\mathbf{v}_1, \mathbf{v}_c$ and $\bar{\mathbf{v}}_1, \mathbf{v}_c$ coincide: the coefficient marked as a disk is on the bisectrix. This fixes the large-circled point in (a). (c) and (d): cubic second derivative layer. (c) Circled point of cubic layer on sector partitioning ray is free. (d) C^2 extensions of linear and quadratic layers define the points labeled $\circ$. The $\circ$ points and C^1 continuity define the points of cubic layers marked by diamonds. The diamonds, filled disks and C^2 continuity define the remaining Bézier points of the cubic layer. The Bézier points of the C^2 extensions of these cubic layers (along $\mathbf{v}_1, \mathbf{v}_c$) coincide: the coefficient marked as a disk is on the bisectrix. This fixes the large-circled point in (c).

the transition map τ along the outline of the V-shape, and, in turn, is used to compute the transition between the central cap and the guided rings.

The tensor-border data, position, first and second derivative, are defined by three layers of Bézier coefficients. The first layer is piecewise linear and the second piecewise quadratic as follows.

(i) Along the polygon edges $(\mathbf{0}, \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_c, \bar{\mathbf{v}}_1, \bar{\mathbf{v}}_0)$, σ is linear, and represented (thin lines in Figure 4, *left*) in degree-raised Bézier form:

$$\sigma_{02} := \left[\begin{smallmatrix} x_3 \\ 0 \end{smallmatrix}\right],\ \sigma_{01} := \left[\begin{smallmatrix} x_1 \\ 0 \end{smallmatrix}\right],\ \sigma_{i0} := (1 - \frac{i}{3})\mathbf{v}_0 + \frac{i}{3}\mathbf{v}_1,\ \sigma_{3j} := (1 - \frac{j}{3})\mathbf{v}_1 + \frac{j}{3}\mathbf{v}_c,$$

with the prolongation $\mathbf{e} := \frac{4}{3}\mathbf{v}_c - \frac{1}{3}\bar{\mathbf{v}}_1$.
(ii) The first, inner layer from the tip is quadratic in degree-raised form (medium lines in Figure 4, *left*). Since pieces corresponding to the same polygon edge join C^2, they are actually a single quadratic and

(iii) Only the third layer (fat lines in Figure 4, *left*) is of actual degree 3 and consists of more than one polynomial piece per polygon edge.
Second and third layer are defined as

$$\sigma_{11} := \left[\begin{smallmatrix} x_1 \\ y_1 \end{smallmatrix}\right],\ \sigma_{21} := \left[\begin{smallmatrix} x_2 \\ y_2 \end{smallmatrix}\right],\ \sigma_{12} := \left[\begin{smallmatrix} x_3 \\ y_3 \end{smallmatrix}\right],\ \sigma_{22} := \left[\begin{smallmatrix} x_4 \\ y_4 \end{smallmatrix}\right].$$

(iv) We now subdivide σ at the midpoints in the u and the v direction to obtain the layout of Figure 4, *right*, and set $\mathbf{m} := \left[\begin{smallmatrix} x_5 \\ y_5 \end{smallmatrix}\right]$. By choosing

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} := \begin{pmatrix} 1 & \frac{-\hat{c}}{2\hat{s}} & \frac{\hat{c}}{6} \\ \frac{8}{9} & \frac{-\hat{c}}{3\hat{s}} & \frac{2\hat{c}}{9} \\ \frac{13-c}{18} & \frac{(c-7)\hat{c}}{18\hat{s}} & \frac{5\hat{c}}{18} \\ \frac{5-c}{9} & \frac{(c-4)\hat{c}}{9\hat{s}} & \frac{2\hat{c}}{3} \\ \frac{189-19c}{288} & \frac{(12c-55)\hat{c}}{192\hat{s}} & \frac{91\hat{c}}{192} \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix},\ \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} := \begin{pmatrix} 0 & \frac{2}{9} & \frac{\hat{s}}{9} \\ 0 & \frac{4}{9} & \frac{2\hat{s}}{9} \\ \frac{-s}{18} & \frac{c+2}{18} & \frac{\hat{s}}{3} \\ \frac{-s}{9} & \frac{c+2}{9} & \frac{2\hat{s}}{3} \\ \frac{-19s}{288} & \frac{9c+47}{144} & \frac{17\hat{s}}{36} \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix},$$

we satisfy the C^2 continuity constraints. We summarize the result in Lemma 1.

Lemma 1. *The tessellation map σ defines C^2 data along $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_c, \bar{\mathbf{v}}_1, \bar{\mathbf{v}}_0)$. With R the rotation by $\frac{2\pi}{n}$, the maps of two adjacent sectors, σ and $R\sigma$, join C^2 at $\bar{\mathbf{v}}_0$.*

Using the explicit formulas, it is easy to formally verify Lemma 1. In particular, C^2 extensions of σ on $\overline{\mathbf{v}_0\mathbf{v}_1}$ and $\overline{\bar{\mathbf{v}}_0\bar{\mathbf{v}}_1}$ join at the *circled* point in Figure 6,*right*. This will allow constructing τ of degree (4,4). The intuition behind the approach and a specific, unique instance of σ are sketched in Figure 5.

3 Transition τ from σ to ρ

We now build a transition τ between the innermost parameterization σ of the V-shape and the outer parameterization ρ, the characteristic map of Catmull-Clark subdivision. One sector of the transition will consist of 16 patches of degree (4,4) as shown in Figure 7. The Bézier net of one of the n rotationally equivalent sectors of τ^*, a precursor of τ, is shown in Figure 6. The outer (*dark grey*) layers of Bézier control points are defined by the C^2 prolongation of ρ and the inner (*light grey*) layers by σ. Then the internal C^2 constraints leave only the control points marked as *black* disks free to choose. Symmetry further cuts the number of free scalar values to 20. For $n = 3$, all 20 free parameters are determined using the functional $\mathcal{F}_4$ below. For $n > 4$, we set $\mathbf{v}_0 := \lambda \mathbf{w}_0$, where λ is subdominant eigenvalue of Catmull-Clark subdivision. The remaining 19 parameters are determined by minimizing the functional $\mathcal{F}_4$ summed over all 16 pieces of transition. Here $\mathcal{F}_k(f_1, f_2) := \mathcal{F}_k(f_1) + \mathcal{F}_k(f_2)$ and

$$\mathcal{F}_k(f) := \int_0^1 \int_0^1 \sum_{i+j=k;\ i,j\geq 0} \binom{k}{i} (\partial_s^i f \partial_t^j f)^2. \tag{1}$$

Note that the functional is not applied to a surface in $\mathbb{R}^3$, but to a map in $\mathbb{R}^2$, and that these maps depend only on the valence, i.e. can be precomputed once and for all. Transition tessellations τ^* for various valences are shown in

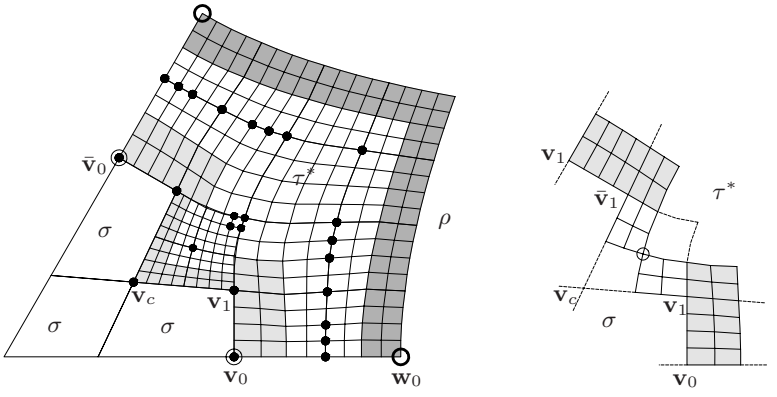


Fig. 6. (*left*) Structure of one sector of the tessellation τ^* between the origin and the outer tessellation ρ. (*right*) The transition τ^* near $\mathbf{v}_c$.

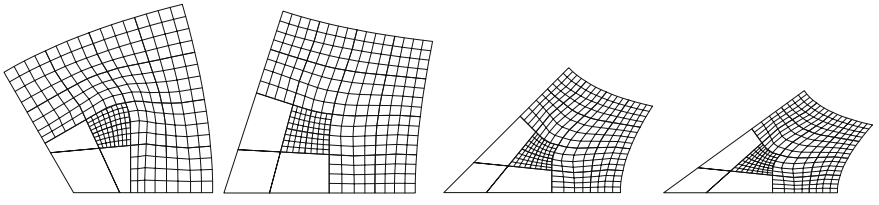


Fig. 7. Bézier net of the transition maps τ^*. (from *left* to *right*): $n = 3, 5, 8, 10$. The case $n = 6$ is shown in Figure 6 *left*.

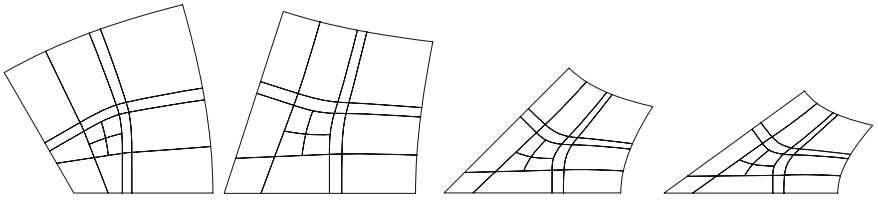


Fig. 8. Polynomial patch boundaries of the transition maps τ with adjusted spacing (from *left* to *right*): $n = 3, 5, 8, 10$

Figure 7. When compared to ρ, the first two derivatives of the transition maps τ^* in Figure 7 are one quarter the size. The univariate example in Figure 9, *right*, indicates that adjusting the spacing to make the derivatives larger improves the transition characteristics. Figure 8 shows the internal partition of the final map τ with the adjusted spacing according to Figure 10, *right*. Explicit formulas for the undetermined Bézier coefficients are provided in the Appendix. The underlying observation is that the change of lengths between τ^* of Figure 7 and ρ causes

problems when the guide is switched at the same time from **g** to **q** (see Figure 11 *left*). In contrast, unequal lengths *within* τ and correspondingly narrow patches sampling from one good guide **q** yields a stable procedure for all valences and does not appear to introduce any geometric artifacts (see Figure 11 *right*).

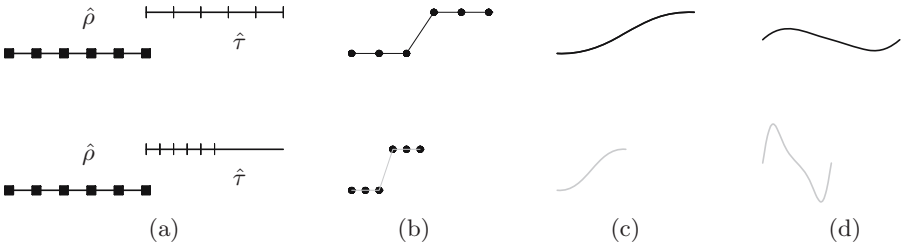


Fig. 9. Univariate example: transition between curves $\hat{\rho}$ and $\hat{\tau}$. (*top*) Long transition. (*bottom*) Short transition. (a) The black squares and partitions indicate two initial Bézier nets, here to be joined by a degree 5 transition. (b) shows the control net of the degree 5 curve (c) and (d) its curvature plot. The *top* transition would typically be preferred.

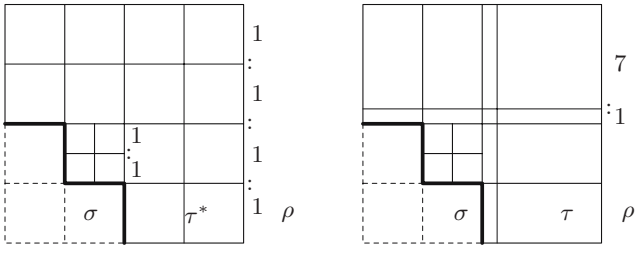


Fig. 10. (*left*) 1:1 spacing of the domain of the map τ^*; (*right*) adjusted transition domain layout for the map τ (the image of the domain under τ was shown in Figure 8)

4 Surface Cap in $\mathbb{R}^3$

The operator H decouples the degree of the composition of guide and tessellation map from the degree of the resulting surface ring. However, for the cap, we need a low-degree guide to arrive at a final surface that has low degree. After three guided surface rings, piecewise quartic guides do well since, in the mth ring, the kth order terms have decayed to λ^{-km} their original size, where $\lambda < 1$ is the subdominant eigenvalue of Catmull-Clark subdivision. So we may replace any initial guide **g** by a C^2 piecewise polynomial guide **q** of total degree $d = 4$.

Standard C^2 prolongation from the last guided ring attaches the transition surface $\mathbf{x}_t := H(\mathbf{q} \circ \tau)$ to the surrounding surface; and $\mathbf{x}_t$ consists of 16 patches of degree (6,6). Figure 11, *left*, illustrates that switching the guide from **g** to **q**. Applying τ^* in the same, third surface ring has a detrimental effect on the curvature

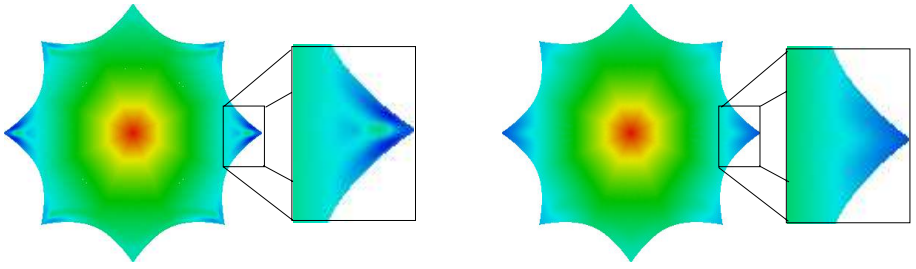


Fig. 11. (*left*) Switching both the tessellation map from ρ to τ^* and the guide from $\mathbf{g}$ to $\mathbf{q}$ in the same ring yields an uneven Gauss curvature distribution (see enlargement). (*right*) Visually identical, better Gauss curvature distribution results from (i) switching to τ^* one ring later or (ii) switching in the same ring, but using the adjusted transition map τ.

distribution. Figure 11, *right*, illustrates two alternative strategies that result in visually identical curvature images. The first switches to $\mathbf{q}$ in the fourth ring and uses τ^* in an additional fifth ring; the second switches guide and tessellation in the same, fourth ring, but uses τ.

The resulting transition patchwork then smoothly joins the innermost cap. Since we only change the parameterization, not the image, the central V-shaped piece can alternatively be parametrized by σ or by β where β consists of three bilinear pieces tracing out the V-shape. For example, one piece has the coefficients $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_c$ and a point on the edge $\mathbf{0}\bar{\mathbf{v}}_0$. That is, the central V-shaped piece of the surface can be defined by the composition $\mathbf{q} \circ \beta$ of degree (4,4) i.e. be reinterpreted as a bilinearly trimmed piece of the guide $\mathbf{q}$.

Theorem 1 (C^2 continuity). *With R the rotation by $2\pi/n$ and β a parameterization of the V-shaped domain consisting of three bilinear pieces,*

(a) $\mathbf{q} \circ \beta$ *and* $\mathbf{q} \circ R\beta$ *join* G^2 *(G^2 capping);*
(b) $\mathbf{q} \circ \beta$ *and* $H(\mathbf{q} \circ \tau)$ *join* G^2 *(G^2 join to data);*
(c) $deg(\mathbf{q} \circ \beta) = (4, 4)$.

Proof. Since β and σ have the same V-shaped range, the point sets $\mathbf{q} \circ \beta$ and $\mathbf{q} \circ \sigma$ agree in $\mathbb{R}^3$ (the same piece of the C^2 guide surface is cut out). Since $\mathbf{q} \circ \sigma$ and $\mathbf{q} \circ R\sigma$ join G^2, (a) follows. For any preimage E of a line segment of the boundary $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_c, \bar{\mathbf{v}}_1, \bar{\mathbf{v}}_0)$ of the V-shape, denote by $\deg_E(f)$ the degree of f restricted to E. Then, by construction, $\deg_E(\partial^j \mathbf{q} \circ \sigma) = 4 + j$ for $j = 0, 1, 2$. In particular, $\deg_E(\partial^j \mathbf{q} \circ \sigma) \leq 6$ and hence $H(\mathbf{q} \circ \tau)$ reproduces $\partial^j(\mathbf{q} \circ \sigma)$ along E. Therefore $\mathbf{q} \circ \sigma$ and $H(\mathbf{q} \circ \tau)$ join C^2. By the agreement of $\mathbf{q} \circ \sigma$ and $\mathbf{q} \circ \beta$ explained earlier, (b) holds. Since $\mathbf{q}$ is of degree 4 and β of degree (1,1), the composition $\mathbf{q} \circ \beta$ is of degree (4,4).

Overall, from the outside towards the central point, the construction yields a sequence of surface rings $H(\mathbf{g} \circ \rho)$ of $3n$ patches of degree (6,6), followed by a transition ring $H(\mathbf{q} \circ \tau)$ of 16 patches of degree (6,6), followed by a cap $\mathbf{q} \circ \beta$ of $3n$ patches of degree (4,4).

5 Discussion

While a number of techniques now exist that automatically create curvature continuous blends [4,5,6,7,8,9,10,11,12,13,14,15], none fully meets the requirements of product-defining outer surfaces in high-end design of CAD/CAM applications. For example, [16] showed that shape deficiencies are intrinsic to all major subdivision surface constructions, no matter how carefully a designer might adjust the input data. Algorithms that build on, say Catmull-Clark, subdivision inherit the problems. Conversely, constructions that do not make use of multiple refinement steps and construct the blend from a minimal number of polynomial pieces typically pay for the abrupt transition of patch type with curvature fluctuations.

In contrast to the prevalent all-in-one approach, the guided approach explicitly separates the initial shape design from surface representation. The initial shape is defined via local surface fragments that need only obey few constraints. These fragments are then sampled (re-approximated) so that the output fits standard design cycles.

Compared to the C^2 scheme in [1], the presented approach has lower overall degree. Compared to [3] the number of patches is halved and the layout is simpler so that the scheme is stable even for higher valences. The curvature distribution of all three schemes is very similar.

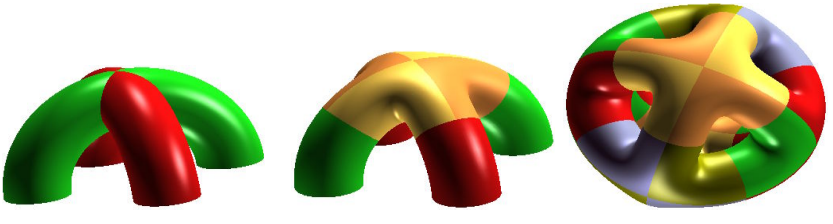


Fig. 12. (*left* and *middle*) Joining two torus pieces with curvature continuity. (*right*) Joining three tori.

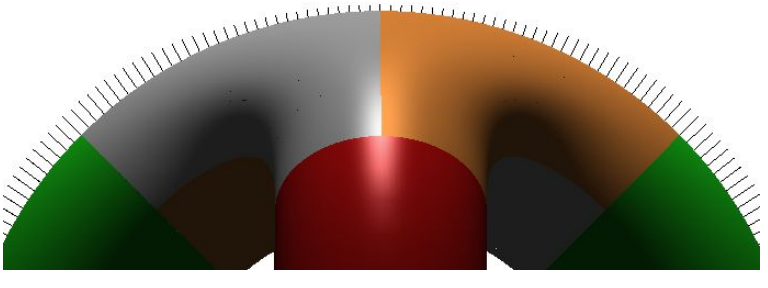


Fig. 13. Curvature comb (normals scaled by Gauss curvature attached to surface points) for the three tori join

Acknowledgements. This work was supported in part by grants DMI-0400214 and CCF-0430891.

References

1. Karčiauskas, K., Peters, J.: Guided C^2 spline surfaces. Computer Aided Geometric Design, (submitted) (2007)
2. Karčiauskas, K., Peters, J.: Concentric tesselation maps and curvature continuous guided surfaces. Computer Aided Geometric Design 24(2), 99–111 (2007)
3. Karčiauskas, K., Peters, J.: Parameterization transition for guided c^2 surfaces of low degree. In: Sixth AFA Conference on Curves and Surfaces Avignon, June 29-July 5, 2006, pp. 183–192 (2007)
4. Peters, J.: C^2 free-form surfaces of degree (3,5). Computer Aided Geometric Design 19(2), 113–126 (2002)
5. Hahn, J.: Filling polygonal holes with rectangular patches. In: Theory and practice of geometric modeling (Blaubeuren, 1988), pp. 81–91. Springer, Berlin (1989)
6. Gregory, J.A., Hahn, J.M.: A C^2 polygonal surface patch. Comput. Aided Geom. Design 6(1), 69–75 (1989)
7. Gregory, J.A., Zhou, J.: Irregular C^2 surface construction using bi-polynomial rectangular patches. Comput. Aided Geom. Design 16(5), 423–435 (1999)
8. Ye, X.: Curvature continuous interpolation of curve meshes. Computer Aided Geometric Design 14(2), 169–190 (1997)
9. Reif, U.: Analyse und Konstruktion von Subdivisions algorithmen für Freiformflächen beliebiger Topologie. Shaker Verlag, Aachen (1999)
10. Prautzsch, H.: Freeform splines. Comput. Aided Geom. Design 14(3), 201–206 (1997)
11. Prautzsch, H., Umlauf, G.: Triangular G^2 splines. In: Laurent, P.J., LeMéhauté, A., Schumaker, L.L. (eds.) Curve and Surface Design, Vanderbilt University Press, pp. 335–342 (2000)
12. Navau, J.C., Garcia, N.P.: Modeling surfaces from meshes of arbitrary topology. Comput. Aided Geom. Design 17(7), 643–671 (2000)
13. Loop, C.: Second order smoothness over extraordinary vertices. In: Symp Geom Processing, pp. 169–178 (2004)
14. Ying, L., Zorin, D.: A simple manifold-based construction of surfaces of arbitrary smoothness. ACM TOG 23(3), 271–275 (2004)
15. Levin, A.: Modified subdivision surfaces with continuous curvature. In: SIGGRAPH, ACM Transactions On Graphics, pp. 1035–1040 (2006)
16. Karciauskas, K., Peters, J., Reif, U.: Shape characterization of subdivision surfaces – case studies. Computer Aided Geometric Design 21(6), 601–614 (2004)

Appendix

The transition map τ is defined by scalar values $[z_0, z_2, \ldots, z_{19}]$ (see Table 1) that define coefficients $\mathbf{p}_1, \ldots \mathbf{p}_{15}$ as indexed in Figure 14. The remaining coefficients are defined by symmetry and the C^2 conditions.

$$\mathbf{p}_1 := \mathbf{v}_0 := (z_0, 0),\ \mathbf{p}_2 := \mathbf{v}_1 := (z_0, z_1),\ \mathbf{p}_3 := \mathbf{v}_2 := (z_2\hat{\mathsf{c}}, z_2\hat{\mathsf{s}}),$$
$$\mathbf{p}_4 := (z_3, 0),\ \mathbf{p}_5 := (z_3, z_4),\ \mathbf{p}_6 := (z_5, 2z_4),\ \mathbf{p}_7 := (z_6, z_7),$$
$$\mathbf{p}_8 := (z_8, z_9),\ \mathbf{p}_9 := (z_{10}, z_{11}),\ \mathbf{p}_{10} := (z_{12}, z_{13}),\ \mathbf{p}_{11} := (z_{14}\hat{\mathsf{c}}, z_{14}\hat{\mathsf{s}}),$$
$$\mathbf{p}_{12} := (z_{15}, z_{16}),\ \mathbf{p}_{13} := (z_{17}\hat{\mathsf{c}}, z_{17}\hat{\mathsf{s}}),\ \mathbf{p}_{14} := (z_{18}\hat{\mathsf{c}}, z_{18}\hat{\mathsf{s}}),\ \mathbf{p}_{15} := (z_{19}\hat{\mathsf{c}}, z_{19}\hat{\mathsf{s}})\ .$$

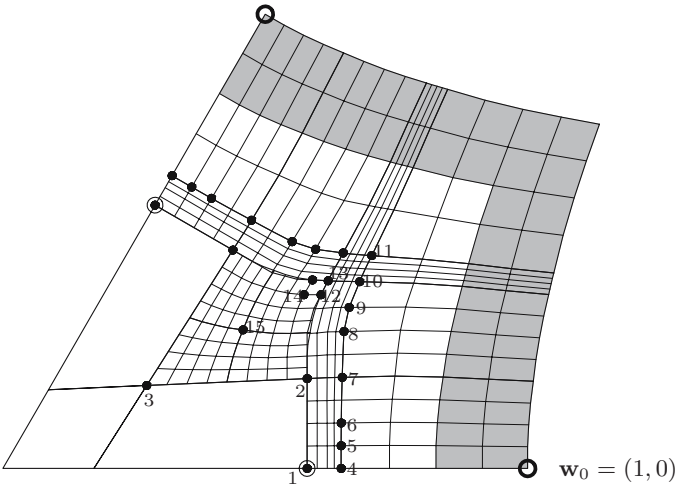


Fig. 14. Indices of the free points $\mathbf{p}_i$ of a half-segment

Table 1. Scalar parameters defining τ for valence n

i of z_i	n=3	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12
0	.34871	.54999	.57968	.59851	.61112	.61994	.62634	.63113	.63480
1	.28552	.20389	.17136	.14718	.12853	.11373	.10173	.09183	.08353
2	.29433	.32590	.31641	.31374	.31414	.31610	.31891	.32217	.32566
3	.41705	.61441	.64457	.66326	.67544	.68368	.68941	.69349	.69643
4	.07309	.05140	.04334	.03727	.03257	.02882	.02578	.02326	.02115
5	.41698	.61450	.64476	.66354	.67580	.68412	.68994	.69409	.69712
6	.41673	.61563	.64672	.66613	.67890	.68767	.69389	.69841	.70178
7	.29215	.20589	.17372	.14945	.13060	.11559	.10339	.09331	.08485
8	.41653	.61752	.64994	.67032	.68385	.69324	.70000	.70500	.70880
9	.43795	.30923	.26108	.22468	.19637	.17382	.15547	.14030	.12757
10	.40992	.62393	.65985	.68238	.69737	.70781	.71536	.72099	.72530
11	.50993	.36235	.30704	.26515	.23255	.20656	.18541	.16790	.15321
12	.39526	.63705	.68004	.70687	.72469	.73712	.74614	.75288	.75807
13	.58072	.41754	.35611	.30937	.27290	.24378	.22006	.20040	.18386
14	.75262	.80556	.81170	.81568	.81840	.82035	.82181	.82294	.82385
15	.33372	.56708	.60617	.63077	.64721	.65873	.66712	.67342	.67828
16	.53111	.38919	.33118	.28738	.25332	.22617	.20407	.18576	.17036
17	.65178	.71175	.71566	.71856	.72071	.72233	.72360	.72462	.72546
18	.60341	.66137	.66243	.66379	.66503	.66607	.66695	.66769	.66832
19	.45588	.52668	.52838	.53095	.53345	.53570	.53769	.53946	.54105

MOS Surfaces: Medial Surface Transforms with Rational Domain Boundaries

Jiří Kosinka and Bert Jüttler

Johannes Kepler University, Institute of Applied Geometry, Altenberger Str. 69,
A–4040 Linz, Austria
{Jiri.Kosinka,Bert.Juettler}@jku.at

Abstract. We consider rational surface patches $\mathbf{s}(u, v)$ in the four-dimensional Minkowski space $\mathbb{R}^{3,1}$, which describe parts of the medial surface (or medial axis) transform of spatial domains. The corresponding segments of the domain boundary are then obtained as the envelopes of the associated two-parameter family of spheres. If the Plücker coordinates of the line at infinity of the (two-dimensional) tangent plane of $\mathbf{s}$ satisfy a sum-of-squares condition, then the two envelope surfaces are shown to be rational surfaces. We characterize these Plücker coordinates and analyze the case, where the medial surface transform is contained in a hyperplane of the four-dimensional Minkowski space.

1 Introduction

Planar and spatial domains can uniquely be represented by their medial axis transform or medial surface transform (in the literature, the term medial axis transform is sometimes used both for two- and three-dimensional domains. In this paper we use medial surface transform for domains in three-dimensional space). Various applications and computational aspects have been studied in Computer Vision, Computational Geometry and Computer Aided Design: see, e.g., [1,4,6,7,11,16,19,21,22] and the references cited therein.

In the spatial case, the medial surface transform of a domain is a set of surface patches (and possibly curve segments, if canal surfaces are present) in 4-dimensional ($xyzr$-) space. Roughly speaking, each point on one of these curves and surfaces represents the center (xyz) and the radius (r) of a sphere which is contained in the domain and touches the boundary in at least two points. This representation covers the structure of the domain and has many computational advantages. For instance, the operation of offsetting is equivalent to adding a constant to the radii, i.e., to a particular translation in the 4-dimensional space. Moreover, self-intersections of inner offsets can be detected by analyzing the radii of the spheres after the translation (see [4] for a discussion of the three-dimensional case).

Rational representations of curves and surfaces (NURBS curves and surfaces) have become a universally accepted standard for Computer Aided Design. However, this class is not closed under the operation of offsetting. As a remedy for

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 245–262, 2007.
© Springer-Verlag Berlin Heidelberg 2007

this problem, the classes of Pythagorean hodograph curves and Pythagorean normal vector surfaces were introduced [12,14,17,18]. These curves and surfaces are characterized by the property that the length of the tangent or normal vector is a (possibly piecewise) rational function of the curve or surface parameters.

The notion of Pythagorean hodograph curves has been extended to the three-dimensional Minkowski space [2,4,5,16], simply by replacing the Euclidean length of the tangent vector (the hodograph) with the length defined by the indefinite inner product in Minkowski space. This led to the class of Minkowski Pythagorean hodograph curves. It is characterized by the property that the Minkowski length of the first derivative vector is a (piecewise) polynomial function of the curve parameter.

As observed in [4,16], Minkowski Pythagorean hodograph curves in the three-dimensional Minkowski space are particularly well suited for representing the medial axis transforms of planar domains. If a segment of the medial axis transform is a Minkowski Pythagorean hodograph curve, then the associated pieces of the domain boundaries are segments of rational curves. This property is then shared by all offsets of the domain boundaries. Motivated by this observation, computational techniques for representing the medial axis transforms of general planar domains by Minkowski Pythagorean hodograph curves have been studied (see, e.g., [15]).

In the present paper, we generalize this idea to the class of two-dimensional surface patches in four-dimensional Minkowski space. These surface patches are used to describe segments of medial surface transforms. They are closely related to bisector surfaces between two given surfaces (which may degenerate into curves), cf. [9,10].

The remainder of the paper is organized as follows. Section 2 summarizes some basic notions and facts concerning four-dimensional Minkowski geometry and medial surface transforms. The next section focuses on the local differential geometry of sheets od medial surface transforms and on the envelope formula for two-parameter families of spheres. This motivates the definition of MOS surfaces in Section 4. The fifth section analyzes the particular cases of MOS surfaces which are contained in hyperplanes. After briefly addressing the general case we conclude this paper.

2 Preliminaries

This section gives a summary of some basic results and concepts concerning Minkowski space $\mathbb{R}^{3,1}$ and medial surface transforms.

2.1 Minkowski Space $\mathbb{R}^{3,1}$ and Homogeneous Coordinates

The four-dimensional *Minkowski space* $\mathbb{R}^{3,1}$ is a four-dimensional real affine space equipped with the indefinite inner product

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^\top G \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3 - u_4 v_4 \tag{1}$$

defined by the matrix

$$G = (G_{i,j})_{i,j=1,2,3,4} = \mathrm{diag}(1, 1, 1, -1), \tag{2}$$

where $\mathbf{u} = (u_1, u_2, u_3, u_4)^\top$, $\mathbf{v} = (v_1, v_2, v_3, v_4)^\top$. The four axes spanned by the vectors $\mathbf{e}_i = (\delta_{i,1}, \delta_{i,2}, \delta_{i,3}, \delta_{i,4})^\top$, $i = 1, \ldots, 4$, will be called the x-, y-, z- and r-axis, respectively.

As the quadratic form associated with G is not definite, the squared norm of a vector, which is defined by $||\mathbf{v}||^2 = \langle \mathbf{v}, \mathbf{v} \rangle$, can be positive, negative or zero. Motivated by the special theory of relativity, one distinguishes three 'causal characters' of vectors: A vector $\mathbf{v}$ is said to be *space-like* if $||\mathbf{v}||^2 > 0$, *time-like* if $||\mathbf{v}||^2 < 0$, and *light-like* (or isotropic) if $||\mathbf{v}||^2 = 0$.

A hyperplane in Minkowski space is called space-, time- or light-like if the restriction of the quadratic form defined by G on this plane is positive definite, indefinite nondegenerate or degenerate, respectively. The type of a hyperplane ρ can be characterized by the Euclidean angle α included between ρ and the xyz hyperplane. For light-like hyperplanes, $\alpha = \frac{\pi}{4}$. Similarly one may distinguish between space-like, time-like and light-like 2-planes and lines.

A linear transform $L : \mathbb{R}^{3,1} \to \mathbb{R}^{3,1}$ is called a *Lorentz transform* if it maintains the Minkowski inner product, i.e. $\langle \mathbf{u}, \mathbf{v} \rangle = \langle L\mathbf{u}, L\mathbf{v} \rangle$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{3,1}$. The set of all Lorentz transforms $\mathcal{L} = O(3, 1)$, along with the composition, forms the *Lorentz group*. If the matrix $K = (k_{i,j})_{i,j=1,2,3,4}$ represents a Lorentz transform, then its column vectors $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ and $\mathbf{k}_4$ satisfy $\langle \mathbf{k}_i, \mathbf{k}_j \rangle = G_{i,j}$, $i, j \in \{1, 2, 3, 4\}$, i.e., they form an orthonormal basis of $\mathbb{R}^{3,1}$.

By introducing points at infinity we obtain the projective closure of the Minkowski space, which will be denoted by $\tilde{\mathbb{R}}^{3,1}$. Its points will be described using standard homogeneous coordinates $\tilde{\mathbf{p}} = (p_0 : p_1 : p_2 : p_3 : p_4)^\top$ satisfying

$$1 : x : y : z : r = p_0 : p_1 : p_2 : p_3 : p_4. \tag{3}$$

Points at infinity (also called ideal points) are then characterized by $p_0 = 0$.

2.2 Medial Surface Transform

Consider a domain $\Omega \in \mathbb{R}^3$. Any sphere contained in Ω is said to be an inscribed sphere. The set of all inscribed spheres is partially ordered with respect to inclusion. An inscribed sphere is said to be maximal if it is not contained in any other inscribed sphere. The *medial surface* of Ω is the locus of all centers of maximal inscribed spheres.

The *medial surface transform* of the domain is obtained by adding the radius information to the medial surface. The medial surface transform consists of all points $(x, y, z, r)^\top$ in $\mathbb{R}^{3,1}$, where $(x, y, z)^\top$ is a point of the medial surface and r is the radius of the maximal inscribed sphere centered at this point.

For a general domain with piecewise smooth boundary, the medial surface and the medial surface transform both consist of several components with different dimensions, see, e.g., [6]. Here we consider only two-dimensional components, which are called *sheets*. They correspond to the centers of inscribed spheres which

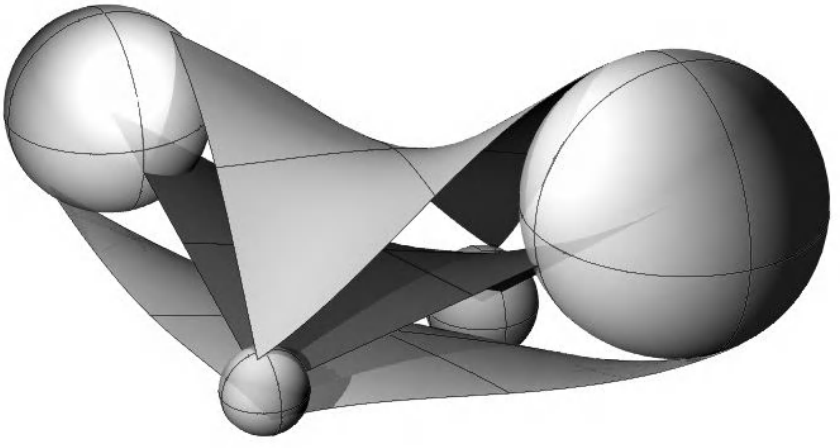


Fig. 1. A medial surface segment (dark grey) and the corresponding pieces of the boundary surface (light grey). In addition, four inscribed spheres are shown.

touch the domain boundary in two points. An example is shown in Fig. 1, which shows a quadrangular segment (patch) of the medial surface, along with the four inscribed spheres at the four vertices and the two corresponding quadrangular patches of the domain boundaries. If

$$\mathbf{s}(u,v) = (x(u,v), y(u,v), z(u,v), r(u,v))^\top, \ (u,v) \in D \subseteq \mathbb{R}^2 \tag{4}$$

is a regular surface patch in $\mathbb{R}^{3,1}$ which describes a sheet of the medial surface transform, then we can recover the associated part of the domain Ω as

$$\Omega = \bigcup_{(u,v)\in D} B_{r(u,v)}(x(u,v), y(u,v), z(u,v)), \tag{5}$$

where $B_r(x,y,z)$ is the ball with center $(x,y,z)^\top$ and radius r. Its boundary $\partial\Omega$ contains the envelope of the spheres $\partial B_{r(u,v)}(x(u,v), y(u,v), z(u,v))$. In addition it contains segments of the canal surfaces which correspond to the boundary curves of $\mathbf{s}$, and segments of the spheres which correspond to the vertices of $\mathbf{s}$.

Remark 1. A two-parameter family of spheres is not always a valid medial surface transform. For instance, the families of spheres can be nested.

3 Tangent Planes of Sheets of Medial Surface Transforms and Envelope Formula

We characterize the tangent 2-planes of sheets of the medial surface transform and derive the formula for computing the two envelope surfaces of the associated 2-parameter sphere family.

3.1 Tangent Planes of 2-Surfaces in $\mathbb{R}^{3,1}$

We consider a C^1 patch $\mathbf{s} : (u, v) \mapsto \mathbf{s}(u, v)$,

$$\mathbf{s}(u, v) = (x(u, v), y(u, v), z(u, v), r(u, v))^\top, \tag{6}$$

of a 2-surface in $\mathbb{R}^{3,1}$, where the parameters (u, v) vary in a suitable parameter domain $D \subset \mathbb{R}^2$. The components of the associated first fundamental form are

$$E = \langle \mathbf{s}_u, \mathbf{s}_u \rangle, \quad F = \langle \mathbf{s}_u, \mathbf{s}_v \rangle, \quad G = \langle \mathbf{s}_v, \mathbf{s}_v \rangle, \tag{7}$$

with the partial derivative vectors[1]

$$\mathbf{s}_u = \frac{\partial \mathbf{s}(u, v)}{\partial u} \quad \text{and} \quad \mathbf{s}_v = \frac{\partial \mathbf{s}(u, v)}{\partial v}. \tag{8}$$

At each regular point of $\mathbf{s}(u, v)$, the *tangent plane* is spanned by the two vectors $\mathbf{s}_u$ and $\mathbf{s}_v$. These vectors span a two-dimensional plane, which is embedded in the four-dimensional Minkowski space $\mathbb{R}^{3,1}$.

The intersection of a tangent 2-plane with the hyperplane ω at infinity defines a line $\nu = \nu(u, v)$. It is spanned by the two points

$$(0 : x_u : y_u : z_u : r_u)^\top \quad \text{and} \quad (0 : x_v : y_v : z_v : r_v)^\top. \tag{9}$$

We describe this line ν in the hyperplane at infinity using Plücker coordinates (see e.g. [20] for an introduction to line geometry):

Definition 1. *For any regular surface patch $\mathbf{s}(u, v)$ in $\mathbb{R}^{3,1}$ we consider the homogeneous coordinates*

$$(p_{01} : p_{02} : p_{03} : p_{23} : p_{31} : p_{12})^\top, \tag{10}$$

where

$$\begin{array}{lll} p_{01} = x_u y_v - y_u x_v, & p_{02} = x_u z_v - z_u x_v, & p_{03} = x_u r_v - r_u x_v, \\ p_{23} = z_u r_v - r_u z_v, & p_{31} = r_u y_v - y_u r_v, & p_{12} = y_u z_v - z_u y_v. \end{array} \tag{11}$$

*They will be called the **Plücker coordinates of the infinite line of the tangent plane**, or **PILT** for short. The vector obtained by collecting them is called the **PILT vector**.*

Remark 2. 1. As for all Plücker coordinates, the six components of the PILT vector are not independent; they satisfy the Plücker condition

$$p_{01} p_{23} + p_{02} p_{31} + p_{03} p_{12} = 0. \tag{12}$$

[1] Throughout the remainder of this paper, the subscripts $*_u, *_v$ denote partial derivatives of scalar or vector-valued quantities.

2. The PILT vector can be seen as a generalization of the normal vector of a surface in three-dimensional Euclidean space. Indeed, the components of the PILT vector are the non-vanishing 3×3 subdeterminants of the matrix

$$S = \begin{pmatrix} 0 & x_u & y_u & z_u & r_u \\ 0 & x_v & y_v & z_v & r_v \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{13}$$

while the components of the normal vector of a surface (X, Y, Z) can be obtained as the non-vanishing subdeterminants of the matrix

$$R = \begin{pmatrix} 0 & X_u & Y_u & Z_u \\ 0 & X_v & Y_v & Z_v \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{14}$$

where the subscripts indicate (again) partial derivatives with respect to the surface parameters.

3.2 Envelope Formula

We consider the two-parameter family of spheres

$$(X - x(u,v))^2 + (Y - y(u,v))^2 + (Z - z(u,v))^2 - r^2(u,v) = 0 \tag{15}$$

which correspond to the C^1 patch (6) of the medial surface transform. In addition to (15), a point $\mathbf{b}$ of one of the two envelopes satisfies the two equations

$$\begin{aligned} (X - x) \cdot x_u + (Y - y) \cdot y_u + (Z - z) \cdot z_u + r \cdot r_u = 0, \\ (X - x) \cdot x_v + (Y - y) \cdot y_v + (Z - z) \cdot z_v + r \cdot r_v = 0. \end{aligned} \tag{16}$$

where x, y, z and r depend on u, v, and $\mathbf{b} = (X, Y, Z)$. Consequently, the coordinates of $\mathbf{b}$ can be found by intersecting the sphere (15) with two planes (16). This leads to a closed-form expression for the two branches $(\pm)$ of the envelope surface:

$$\mathbf{b}^{\pm}(u,v) = \hat{\mathbf{s}}(u,v) + \frac{r}{E_0 G_0 - F_0^2} \left(\mathbf{w} \pm \sqrt{EG - F^2} \cdot (\hat{\mathbf{s}}_u \times \hat{\mathbf{s}}_v) \right). \tag{17}$$

In this expression,

$$\hat{\mathbf{s}}(u,v) = (x(u,v), y(u,v), z(u,v))^{\top} \tag{18}$$

represents the corresponding medial surface, which is obtained by omitting the radius function in (6).

The components E, F, G of the first fundamental form of $\mathbf{s}(u,v)$, see (7), are computed using the indefinite inner product (1) in $\mathbb{R}^{3,1}$. In addition, E_0, F_0, G_0 are the components of the first fundamental form of the corresponding medial surface (18), which are obtained by using the standard Euclidean inner product.

The coordinates of the vector $\mathbf{w} = \mathbf{w}(u, v)$ in (17) are polynomials of degree four in x_u, y_u, z_u, r_u and x_v, y_v, z_v, r_v, where each coordinate consists of 8 terms. Also the other terms are quite complicated. For instance, the squared Minkowski area element $EG - F^2$ in (17) consists of 18 terms of degree 4 in the partial derivatives of x, y, z and r.

A more compact representation can be derived by using the components of the PILT vector of the patch $\mathbf{s}(u, v)$ of the medial surface transform.

Lemma 1. *The two envelope surfaces (17) of the two-parameter family of spheres corresponding to (6) can be rewritten as*

$$\mathbf{b}^{\pm}(u,v) = \hat{\mathbf{s}} + \frac{r}{p_{01}^2 + p_{02}^2 + p_{12}^2} \left(\begin{pmatrix} p_{02}p_{23} - p_{01}p_{31} \\ p_{23}p_{12} - p_{01}p_{03} \\ p_{31}p_{12} - p_{02}p_{03} \end{pmatrix} \pm \sqrt{C} \cdot \begin{pmatrix} p_{12} \\ -p_{02} \\ p_{01} \end{pmatrix} \right), \quad (19)$$

where

$$C = EG - F^2 = p_{01}^2 + p_{02}^2 - p_{03}^2 - p_{23}^2 - p_{31}^2 + p_{12}^2. \quad (20)$$

This observation can be verified by a direct computation.

Remark 3.

1. If the PILT vector vanishes, then the corresponding point of the medial surface transform is singular. Otherwise it is regular, i.e., $\mathbf{s}_u$ and $\mathbf{s}_v$ are linearly independent.
2. If $C = EG - F^2$ is positive, then the tangent 2-plane is space-like and the surface $\mathbf{s}(u, v)$ is regular. Moreover, since $p_{01}^2 + p_{02}^2 + p_{12}^2 > C$, the medial surface $\hat{\mathbf{s}}(u, v)$ is regular, too. Both envelope surfaces (19) are then well-defined.
3. If $C = EG - F^2 = 0$ and $\mathbf{s}$ is regular, then the 2-surface $\mathbf{s}$ has a light-like tangent 2-plane. Due to the regularity of $\mathbf{s}$, not all components of the PILT vector vanish. Consequently, the medial surface is still regular, since

$$E_0 G_0 - F_0^2 = p_{01}^2 + p_{02}^2 + p_{12}^2. \quad (21)$$

   In this case, the envelope formula yields one point only.
4. If $C = EG - F^2$ is negative, then $\mathbf{b}^{\pm}$ is not defined. This characterizes the situation where the tangent 2-plane is time-like.
5. Medial surface transforms with well defined envelope surface(s) will be called regular. From now on, we consider only regular medial surface transforms. These are medial surface transforms which have non-vanishing PILT vectors and either light-like or space-like tangent 2-planes only. This generalizes similar conditions for the medial axis transform of planar domains, which can be identified with a system of curves in Minkowski space $\mathbb{R}^{2,1}$. In that situation, curves with time-like tangents have to be excluded.
6. Lorentz transforms preserve the regularity of the PILT vector and the causal character of the tangent 2-plane.

4 Sheets of Medial Surface Transforms Defining Rational Envelopes

In order to obtain rational envelopes, we introduce the notion of MOS surfaces and characterize their PILT vectors.

4.1 MOS Surfaces

Consider a medial surface transform (6), which is described by piecewise rational functions x, y, z and r. We are particularly interested in the situation where the two envelope surfaces obtained from (19) are rational. This motivates the following definition:

Definition 2. *Let* $\mathbf{s}(u,v) = (x(u,v), y(u,v), z(u,v), r(u,v))^\top$ *be a regular polynomial or rational medial surface transform in* $\mathbb{R}^{3,1}$. *We say that* $\mathbf{s}(u,v)$ *is a polynomial or rational* ***medial surface transform which obeys the sum-of-squares-condition (MOS surface*** *for short), if there exists a bivariate polynomial or rational function* $\sigma(u,v)$, *such that the components of the PILT vector satisfy*

$$p_{01}^2 + p_{02}^2 - p_{03}^2 - p_{23}^2 - p_{31}^2 + p_{12}^2 = \sigma(u,v)^2. \tag{22}$$

MOS surfaces generalize the notion of Minkowski Pythagorean hodograph curves, see [4,16]. If a segment of the medial axis transform of a planar domain is an Minkowski Pythagorean hodograph curve, then the corresponding segments of the domain boundary admit rational parameterization. Moreover, since the translations in Minkowski space preserve the hodographs (first derivative vectors), this is also the case for all offsets of the domain boundaries. In the surface case we get the following result.

Proposition 1. *If a segment of the medial surface transform of a spatial domain is an MOS surface patch, then both the corresponding pieces of the boundary of the domain and their offsets admit a rational parametric representation.*

Remark 4. Using the first fundamental form of $\mathbf{s}(u,v)$, the MOS condition in Definition 2 can be rewritten as

$$EG - F^2 = \langle \mathbf{s}_u, \mathbf{s}_u \rangle \cdot \langle \mathbf{s}_v, \mathbf{s}_v \rangle - \langle \mathbf{s}_u, \mathbf{s}_v \rangle^2 = \sigma(u,v)^2. \tag{23}$$

Consequently, the MOS property is invariant with respect to Lorentz transforms. Moreover, the area (in the Minkowski sense) of a polynomial MOS surface is a (possibly piecewise) polynomial function of the surface parameters u, v.

4.2 Characterizing the PILT Vectors of MOS Surfaces

The 2-surface $\mathbf{s}(u,v)$ in $\mathbb{R}^{3,1}$ has the MOS property if its PILT vector simultaneously satisfies the MOS condition (22) and the Plücker condition (12).

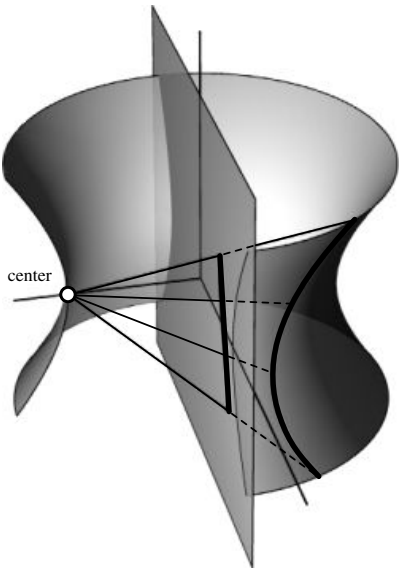


Fig. 2. A three-dimensional sketch of the central projection in $\mathbb{R}^6$. Rational curves in the (hyper)plane are projected onto the (hyper)surface.

Consequently, in order to characterize MOS surfaces, we have to solve a system of two quadratic equations. First, we use a central (or inverse stereographic) projection to solve (22). Let

$$(p_0 : p_1 : p_2 : p_3 : p_4 : p_5 : p_6)^\top, \ p_0 \neq 0, \tag{24}$$

be the homogeneous coordinates of a point in $\mathbb{R}^6$. The center of the projection is chosen at $(1 : -1 : 0 : 0 : 0 : 0 : 0)^\top$. We project points of the hyperplane $p_1 = 0$ onto the quadric hypersurface (22), see Fig. 2 for a sketch. This defines the mapping

$$\begin{pmatrix} p_0 \\ 0 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix} \mapsto \begin{pmatrix} p_0^2 + p_2^2 - p_3^2 - p_4^2 - p_5^2 + p_6^2 \\ p_0^2 - p_2^2 + p_3^2 + p_4^2 + p_5^2 - p_6^2 \\ 2p_0p_2 \\ 2p_0p_3 \\ 2p_0p_4 \\ 2p_0p_5 \\ 2p_0p_6 \end{pmatrix}. \tag{25}$$

Now we substitute the image (25) into Plücker's condition (12), which leads to

$$p_0(-p_4p_2^2 + p_4p_3^2 + p_4^3 + p_4p_5^2 - p_4p_6^2 + p_0^2p_4 + 2p_0p_2p_5 + 2p_0p_3p_6) = 0. \tag{26}$$

This splits into the equation of a cubic hypersurface $\mathcal{C}$ and the hyperplane at infinity, both in $\mathbb{R}^5$. Since $p_0 = 0$ corresponds to the points at infinity, we omit

the factor p_0 in (26). As $\mathcal{C}$ has double points, we use again a central projection in order to obtain a parameterization.

Let $(r_0 : r_1 : r_2 : r_3 : r_4 : r_5)^\top$ be the homogeneous coordinates of a point in $\mathbb{R}^5$. The surface $\mathcal{C}$ has a double point at $(0 : 0 : 0 : 0 : 1 : 1)^\top$. We choose this point as the center of projection and map the points of the hyperplane determined by $r_4 = 0$ onto the cubic hypersurface given by the second factor in (26). A straightforward computation gives the cubic parameterization

$$\begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ 0 \\ r_5 \end{pmatrix} \mapsto \begin{pmatrix} -2(r_0r_1 + r_0r_2 - r_3r_5)r_0 \\ -2(r_0r_1 + r_0r_2 - r_3r_5)r_1 \\ -2(r_0r_1 + r_0r_2 - r_3r_5)r_2 \\ -2(r_0r_1 + r_0r_2 - r_3r_5)r_3 \\ r_3r_0^2 - r_3r_1^2 + r_3r_2^2 + r_3^3 + 2r_0r_2r_5 - r_3r_5^2 \\ r_3r_0^2 - r_3r_1^2 + r_3r_2^2 + r_3^3 + r_3r_5^2 - 2r_5r_0r_1 \end{pmatrix} \tag{27}$$

of $\mathcal{C}$. Finally we combine the parameterization (27) with the projection (25) and obtain (after omitting the common factor $4r_0(r_0r_1+r_0r_2-r_3r_5)$) the final result:

Proposition 2. *If $\mathbf{s}(u, v)$ is an MOS surface, then there exist five polynomials $r_i(u, v)$, $i \in \{0, 1, 2, 3, 5\}$ and a polynomial $\lambda(u, v) \not\equiv 0$ such that the PILT vector of $\mathbf{s}(u, v)$ has the form*

$$\frac{1}{\lambda} \cdot \begin{pmatrix} r_1(r_0^2 - r_1^2 + r_3^2 - r_5^2 - r_1r_2) + r_2(r_0^2 + r_2^2 + r_3^2 + r_5^2 + r_1r_2) \\ 2(r_0r_1 + r_0r_2 - r_3r_5)r_1 \\ 2(r_0r_1 + r_0r_2 - r_3r_5)r_2 \\ 2(r_0r_1 + r_0r_2 - r_3r_5)r_3 \\ -r_3r_0^2 + r_3r_1^2 - r_3r_2^2 - r_3^3 + r_3r_5^2 - 2r_0r_2r_5 \\ -r_3r_0^2 + r_3r_1^2 - r_3r_2^2 - r_3^3 - r_3r_5^2 + 2r_0r_1r_5 \end{pmatrix}. \tag{28}$$

Indeed, the two central projections used for the parameterization are birational mappings, hence any given PILT vector satisfying (12) and (22) can be reproduced, but possibly with extraneous factors (which are taken care of by λ).

5 MOS Surfaces in Hyperplanes

In general, MOS surfaces span the entire Minkowski space $\mathbb{R}^{3,1}$. As a special case, however, an MOS surface can be contained in a hyperplane $\mathcal{H}$. This case will now be discussed in more detail.

5.1 Classification

The hyperplane $\mathcal{H}$ can be space-like, time-like or light-like. Using translations and Lorentz transforms, the hyperplane $\mathcal{H}$ can be mapped to one of the following three standard positions:

$$\begin{aligned} &\mathcal{H} \text{ is space-like} : r = 0, \\ &\mathcal{H} \text{ is time-like} \ : z = 0, \\ &\mathcal{H} \text{ is light-like} \ : z - r = 0. \end{aligned} \tag{29}$$

These three cases will be analyzed separately. We assume that the hyperplane $\mathcal{H}$ has the corresponding standard position.

Case 1: $\mathcal{H}$ is space-like. Since it lies in the hyperplane $r = 0$, the equation of the MOS surface takes the form

$$\mathbf{s}(u,v) = (x(u,v), y(u,v), z(u,v), 0)^\top, \tag{30}$$

i.e. it is contained in the Euclidean 3-space $\mathbb{R}^3$, which is embedded into $\mathbb{R}^{3,1}$. The three components p_{03}, p_{31}, p_{23} of the PILT vector vanish. The remaining components are equal to the components of the cross product $\hat{\mathbf{s}}_u \times \hat{\mathbf{s}}_v$ of the medial surface (18). The MOS condition (22) simplifies to

$$p_{01}^2 + p_{02}^2 + p_{12}^2 = \sigma(u,v)^2. \tag{31}$$

Consequently, these surfaces are PN (Pythagorean normal) surfaces as introduced in [18]. These surfaces are equipped with a rational field of unit normals. Therefore, the offset surfaces are again rational.

In the standard position, both envelope surfaces (17) degenerate into the medial surface. If the MOS surface lies in a plane of the form $r = c$, where c is a constant, then the two envelopes are the offset surfaces at distance c. MOS surfaces in general space-like hyperplanes can be obtained by first embedding PN surfaces in the hyperplane $\mathcal{H}$ of $\mathbb{R}^{3,1}$ and then applying Lorentz transforms.

Case 2: $\mathcal{H}$ is time-like. Since it lies in the hyperplane $z = 0$, the equation of the MOS surface takes the form

$$\mathbf{s}(u,v) = (x(u,v), y(u,v), 0, r(u,v))^\top, \tag{32}$$

i.e. it is contained in the Minkowski 3-space $\mathbb{R}^{2,1}$, which is embedded into $\mathbb{R}^{3,1}$. The three components p_{02}, p_{12}, p_{23} of the PILT vector vanish. The remaining three components are equal to the coordinates of the Minkowski cross product[2] $\tilde{\mathbf{s}}_u \bowtie \tilde{\mathbf{s}}_v$, where

$$\tilde{\mathbf{s}} = (x(u,v), y(u,v), r(u,v))^\top. \tag{33}$$

The MOS condition (22) simplifies to

$$p_{03}^2 + p_{31}^2 - p_{01}^2 = ||\tilde{\mathbf{s}}_u \bowtie \tilde{\mathbf{s}}_v||^2 = -\sigma(u,v)^2. \tag{34}$$

Consequently, the Minkowski norm of the Minkowski cross product $\tilde{\mathbf{s}}_u \bowtie \tilde{\mathbf{s}}_v$ is minus the square of a polynomial. The vector $\tilde{\mathbf{s}}_u \bowtie \tilde{\mathbf{s}}_v$ is therefore time-like.

The medial surface of the MOS surface (32) is always contained in the plane $z = 0$. Consequently, the two envelopes of the associated sphere family are symmetric with respect to the plane $z = 0$.

MOS surfaces in general time-like hyperplanes can be obtained by applying Lorentz transforms to an MOS surface of the form (32). These Lorentz transforms do not preserve the planarity of the medial surface.

[2] $(u_1, u_2, u_3)^\top \bowtie (v_1, v_2, v_3)^\top = (u_2 v_3 - u_3 v_2, u_3 v_1 - u_1 v_3, -u_1 v_2 + u_2 v_1)^\top$, cf. [15].

Case 3: $\mathcal{H}$ is light-like. In the standard position, the equation of the MOS surface takes the form

$$\mathbf{s}(u,v) = (x(u,v), y(u,v), z(u,v), z(u,v))^{\top}. \tag{35}$$

Consequently, in this case, $p_{23} = 0$, $p_{02} = p_{03}$ and $p_{12} = -p_{31}$. The MOS condition simplifies to $p_{01}^2 = \sigma(u,v)^2$. Thus, any rational 2-surface lying in a light-like hyperplane is an MOS surface. One of the two envelopes of the associated family of spheres is then contained in the plane which is obtained by intersecting the light-like hyperplane with the hyperplane $r = 0$, where the latter one is identified with the Euclidean space.

5.2 Construction

MOS surfaces in space-like or time-like hyperplanes $\mathcal{H}$ can be generated with the help of the following construction.

1. Choose a suitable field $\mathbf{N}(u,v)$ of normal vectors. More precisely, find three bivariate polynomials p_{01}, p_{02}, p_{12} or p_{03}, p_{31}, p_{01} in u, v, such that the conditions (31) or (34) are satisfied, respectively. These polynomials can be found by parameterizing the corresponding quadrics. This can be done, e.g., by stereographic projection, or using the method of generalized stereographic projection [8], or its equivalent formulation via Pythagorean hodograph representation maps [5].
2. Find a polynomial or rational surface $\mathbf{s}$ in the standard position of the given hyperplane $\mathcal{H}$, such that the first partial derivatives satisfy

$$\langle \mathbf{N}(u,v), \mathbf{s}_u(u,v) \rangle \equiv 0, \quad \langle \mathbf{N}(u,v), \mathbf{s}_v(u,v) \rangle \equiv 0 \tag{36}$$

   where the inner product $\langle ., . \rangle$ is obtained by restricting the inner product of $\mathbb{R}^{3,1}$ to the standard position of $\mathcal{H}$. If these two conditions are satisfied, then the two vector fields $\mathbf{N}(u,v)$ and $\mathbf{s}_u(u,v) \times \mathbf{s}_v(u,v)$ are linearly dependent for all u, v, and the latter one is therefore obtained by multiplying $\mathbf{N}(u,v)$ by a rational function.
   - A *polynomial* surface satisfying (36) can be obtained by comparing the coefficients. If $\mathbf{N}$ has degree k and $\mathbf{s}$ has degree n in u, v, then the two equations lead to $2\binom{n+k+1}{2}$ equations for the $3\binom{n+2}{2} - 3$ coefficients of $\mathbf{s}$. If n is sufficiently large, then the number of unknowns exceeds the number of linear equations in the homogeneous linear system and nontrivial solutions are therefore guaranteed to exist (for $k = 2$ is $n = 5$ sufficiently large).
   - A *rational* surface satisfying (36) can be generated by computing the envelope of the two-parameter family of planes

$$G(u,v;\mathbf{x}) = \langle \mathbf{N}(u,v), \mathbf{x} \rangle - h(u,v) = 0, \tag{37}$$

     where h(u,v) is another rational function. It can be found by solving the three linear equations $G = G_u = G_v = 0$ for $\mathbf{x}$. This approach has been used in [18] for the construction of PN surfaces.

3. Finally, use a Lorentz transform, in order to map the standard position of the hyperplane into the given general position $\mathcal{H}$.

5.3 Example

We present an MOS surface which is contained in a time-like hyperplane.

Example 1. In step 1, we choose the normal vectors

$$\mathbf{N}(u,v) = (2uv, -2u, u^2 + 1 + v^2)^\top, \tag{38}$$

which satisfy

$$-\mathbf{N}_1^2 - \mathbf{N}_2^2 + \mathbf{N}_3^2 = (u^2 - v^2 - 1)^2. \tag{39}$$

In step 2, in order to generate a polynomial surface, we choose $n = 3$. By comparing the coefficients we obtain a homogeneous system of 42 linear equations for the 42 unknown coefficients, which does not have maximal rank. A particular solution is the polynomial MOS surface

$$\mathbf{s}(u,v) = \begin{pmatrix} -15uv^2 + u^3 - 6uv + \frac{3}{2}v^2 - 3u + 2v \\ 3u^3v - 9uv^3 - 3uv^2 + v^3 + u^3 + v^2 + 3uv + 3u \\ 0 \\ -6u^2v - 3u^2 + 1 \end{pmatrix}. \tag{40}$$

Fig. 3a shows the medial surface and the associated rational domain boundaries. Finally, in step 3, we apply the Lorentz transform

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cosh\beta & \sinh\beta \\ 0 & 0 & \sinh\beta & \cosh\beta \end{pmatrix}, \tag{41}$$

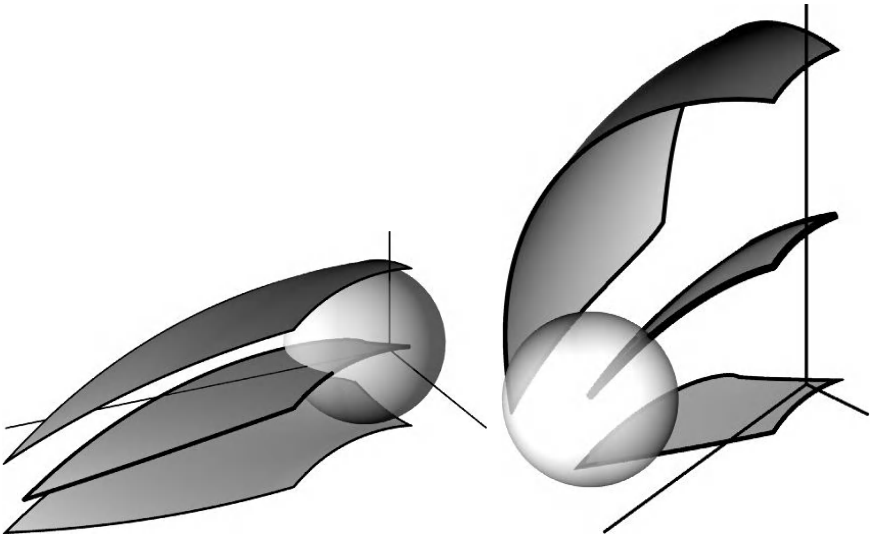
with $\beta = \ln 5$. This leads to the polynomial MOS surface

$$L\,\mathbf{s}(u,v) = \begin{pmatrix} -15v^2u + u^3 - 6vu + \frac{3}{2}v^2 - 3u + 2v \\ 3vu^3 - 9v^3u - 3v^2u + v^3 + u^3 + v^2 + 3vu + 3u \\ \frac{1}{5}(-72vu^2 - 36u^2 + 12) \\ \frac{1}{5}(-78vu^2 - 39u^2 + 13) \end{pmatrix} \tag{42}$$

in the time-like hyperplane $13z = 12r$, see Fig. 3b. It corresponds to two rational domain boundaries with a non-planar medial (bisector) surface.

6 General MOS Surfaces

While MOS surfaces lying in hyperplanes are now fully understood, the situation is less clear in the general case.



(a) (b)

Fig. 3. Example 1: The MOS surface $\mathbf{s}(u, v)$ in the time-like plane $z = 0$ (a) and the same MOS surface after applying the Lorentz transform (41). In both cases, one of the 'corner' spheres is also shown.

6.1 Existence and Examples

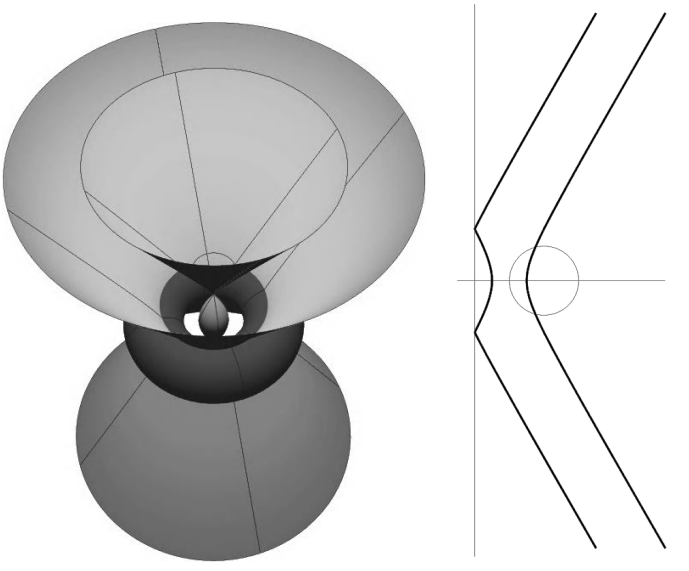
Still we have the following result.

Proposition 3. *There exist both polynomial and rational MOS surfaces.*

Proof. Consider a non-planar polynomial Minkowski Pythagorean hodograph curve in $\mathbb{R}^{2,1}$ and the associated planar domain Ω_0 in the xy-plane. Assume that the y-axis does not intersect Ω_0. Then the spatial domain obtained by rotating Ω_0 around the y-axis has an medial surface transform which can be described by a rational MOS surface, and the spatial domain obtained by translating Ω_0 parallel to the z-axis has an medial surface transform which can be described by a polynomial MOS surface. □

Example 2. Consider a torus and a point $\mathbf{S}$ in $\mathbb{R}^3$. Their bisector surface, along with the associated radius function, defines a rational MOS surface. For the sake of simplicity, we choose the radii of the torus to be 1 and 2, respectively. Moreover, we assume that the center of the torus is at $\mathbf{S} = (0, 0, 0)^\top$ (see Fig. 4). A rational parameterization of the torus is

$$\mathbf{T}(u, v) = \left(\frac{-(3 + v^2)(-1 + u^2)}{(1 + v^2)(1 + u^2)}, \frac{2(3 + v^2)u}{(1 + v^2)(1 + u^2)}, \frac{2v}{(1 + v^2)} \right)^\top . \tag{43}$$



a) b)

Fig. 4. Example 2: Bisector surface of a torus (dark) and a point (center of the torus). a) 3D view, b) The half-meridians of the bisector surface and of the torus.

Using the method described in [10] we compute the rational parameterization of the bisector $\hat{\mathbf{s}}(u,v)$ by solving the following system of linear equations for the coordinates of the bisector surface,

$$\begin{aligned} &\langle \hat{\mathbf{s}}(u,v) - \mathbf{T}(u,v), \mathbf{T}_u \rangle_E = 0, \\ &\langle \hat{\mathbf{s}}(u,v) - \mathbf{T}(u,v), \mathbf{T}_v \rangle_E = 0, \\ &2\langle \hat{\mathbf{s}}(u,v), \mathbf{T}(u,v) - \mathbf{S} \rangle_E = \langle \mathbf{T}(u,v), \mathbf{T}(u,v) \rangle_E - \langle \mathbf{S}, \mathbf{S} \rangle_E. \end{aligned} \tag{44}$$

Here, $\langle ., . \rangle_E$ denotes the Euclidean inner product. The bisector surface is

$$\hat{\mathbf{s}}(u,v) = \left(\frac{(1-u^2)(-9+v^2)}{2(1+u^2)(v^2-3)}, \frac{u(-9+v^2)}{(1+u^2)(v^2-3)}, \frac{3v}{(v^2-3)} \right)^\top. \tag{45}$$

It is a surface of revolution around the z-axis. This surface and its meridian curve is shown in Figure 4. The corresponding rational radius function can be computed by evaluating the Euclidean distance

$$||\hat{\mathbf{s}}(u,v) - \mathbf{S}||_E = \frac{9+v^2}{2(v^2-3)}. \tag{46}$$

This leads to the medial surface transform

$$\mathbf{s}(u,v) = \left(\frac{(1-u^2)(-9+v^2)}{2(1+u^2)(v^2-3)}, \frac{u(-9+v^2)}{(1+u^2)(v^2-3)}, \frac{3v}{v^2-3}, \frac{9+v^2}{2(v^2-3)} \right)^\top. \tag{47}$$

In order to verify that this is indeed an MOS surface, we evaluate

$$\langle \mathbf{s}_u, \mathbf{s}_u \rangle \cdot \langle \mathbf{s}_v, \mathbf{s}_v \rangle - \langle \mathbf{s}_u, \mathbf{s}_v \rangle^2 = \frac{9(v-3)^2(v+3)^2}{(1+u^2)^2(v^2-3)^4} \tag{48}$$

and obtain the square of a rational function. Finally, we show that $\mathbf{s}(u, v)$ is not contained in a hyperplane. Indeed, the five points of $\mathbf{s}(u, v)$ with the parameter values

$$(u, v) \in \{(-1, 0), (-1, 1), (0, 1), (1, -1), (1, 1)\} \tag{49}$$

can be shown to be in general position.

For the sake of simplicity we chose a special mutual position of the torus and the point. However, it can be shown that for any radii of the torus and any position of the point $\mathbf{S}$, the bisector surface along with the associated radius function defines an MOS surface. Since offsetting corresponds to a translation in Minkowski space, this is also true for sphere/torus bisectors. By applying general Lorentz transforms, this even applies to bisectors of certain cyclides and spheres.

6.2 Towards the Construction of General MOS Surfaces

If the tangent 2-planes of two surfaces $\mathbf{s} = (x, y, z, r)$ and $\mathbf{s}' = (x', y', z', r')$, which describe medial surface transforms of spatial domains, share the same line at infinity, then the two 4×3 matrices

$$(\mathbf{s}_u, \mathbf{s}'_u, \mathbf{s}'_v) \quad \text{and} \quad (\mathbf{s}_v, \mathbf{s}'_u, \mathbf{s}'_v) \tag{50}$$

have both rank 2. By evaluating the 3×3 subdeterminants, this can be characterized by two times two equations, which are linear in the partial derivatives of $\mathbf{s}$. These equations involve the components of the PILT vector of $\mathbf{s}'$, since this vector consists of the 2×2 subdeterminants of the last two columns.

This may potentially be useful for constructing MOS surfaces with a given field of PILT vectors. Recall that these vectors of general MOS surfaces can be obtained from Proposition 2. Once a PILT vector has been found, one may try to find a matching MOS surface. In order to find a polynomial MOS surface which is compatible with a given PILT vector one may generate a system of linear equations by comparing the coefficients.

A similar approach may be used for rational MOS surfaces, leading to a system of non-linear equations.

However, even in the polynomial case, the number of equations exceeds the number of free parameters, and solutions do not exist in general. This is due to the fact that the number of equations grows roughly with the same speed (it is described by a quadratic polynomial in n with the leading term $4n^2$, where n is the degree of the unknown MOS surface) as the number of free parameters (it is also a polynomial in n with the same leading term $4n^2$).

This is a major difference to the hyperplanar case. In that case, the number of equations is described by a polynomial with the leading term $2n^2$, while the number of free parameters is a polynomial with the leading term $3n^2$. Consequently, if n is large enough, then more free parameters than equations are present. Unfortunately, this argument does not apply to the general case.

7 Conclusion

Minkowski Pythagorean hodograph curves can be used to represent the medial axis transform of planar domains [4,15,16]. In this paper we generalized this notion to the case of surfaces. We introduced the new class of MOS surfaces, which are special 2-surfaces in the four-dimensional Minkowski space $\mathbb{R}^{3,1}$. By using them to describe segments of the medial surface transform one obtains rational segments of the associated domain boundary.

MOS surfaces can be characterized by the so-called PILT vector, which represents the direction of the tangent 2-plane. Using this vector provides a particularly compact representation of the envelope formula.

The case of MOS surfaces in hyperplanes was studied in some detail. MOS surfaces in space-like planes are equivalent to PN surfaces, see [17,18]. MOS surfaces in time-like planes seem to possess no counterpart in the existing literature.

The detailed investigation of MOS surfaces poses many challenging problems, such as the construction of general MOS surfaces. This may lead to a systematic analysis of curve and surface pairs with rational bisector surfaces (cf. [9,10]), where the associated radius functions are rational too.

Acknowledgment

The authors were supported by the Austrian Science Fund (FWF) through project P17387-N12. Special thanks go to Josef Schicho for his help with the parameterization of the PILT vectors of MOS surfaces. The authors would like to thank the referees for their comments which have helped to improve the first version of this paper.

References

1. Amenta, N., Choi, S., Kolluri, R.K.: The Power Crust, Unions of Balls, and the Medial Axis Transform. Comput. Geom. 19, 127–153 (2001)
2. Cho, H.Ch., Choi, H.I., Kwon, S.-H., Lee, D.S., Wee, N.-S.: Clifford Algebra, Lorentzian Geometry and Rational Parametrization of Canal Surfaces. Computer Aided Geometric Design 21, 327–339 (2004)
3. Choi, H.I., Choi, S.W., Moon, H.P.: Mathematical Theory of Medial Axis Transform. Pacific J. Math. 181, 57–88 (1997)
4. Choi, H.I., Han, Ch.Y., Moon, H.P., Roh, K.H., Wee, N.S.: Medial Axis Transform and Offset Curves by Minkowski Pythagorean Hodograph Curves. Computer-Aided Design 31, 59–72 (1999)
5. Choi, H.I., Lee, D.S., Moon, H.P.: Clifford Algebra, Spin Representation and Rational Parameterization of Curves and Surfaces. Adv. Computational Mathematics 17, 5–48 (2002)
6. Culver, T., Keyser, J., Manocha, D.: Exact Computation of the Medial Axis of a Polyhedron. Comput. Aided Geom. Des. 21, 65–98 (2004)
7. Dey, T.K., Zhao, W.: Approximating the Medial Axis from the Voronoi Diagram with a Convergence Guarantee. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 387–398. Springer, Heidelberg (2002)

8. Dietz, R., Hoschek, J., Jüttler, B.: An Algebraic Approach to Curves and Surfaces on the Sphere and on Other Quadrics. Comput. Aided Geom. Des. 10, 211–229 (1993)
9. Elber, G., Kim, M.–S.: The Bisector Surface of Rational Space Curves. ACM Transactions on Graphics 17, 32–49 (1998)
10. Elber, G., Kim, M.–S.: Computing Rational Bisectors of Point/Surface and Sphere/Surface Pairs. IEEE Comput. Graph. Appl. 19, 76–81 (1999)
11. Giblin, P., Kimia, B.B.: Transitions of the 3D Medial Axis under a One-parameter Family of Deformations. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 718–734. Springer, Heidelberg (2002)
12. Farouki, R.T.: Pythagorean-hodograph Curves. In: Hoschek, J., Farin, G., Kim, M.-S. (eds.) Handbook of Computer Aided Geometric Design, pp. 405–427. Elsevier, North-Holland (2002)
13. Farouki, R.T., al-Kandari, M., Sakkalis, T.: Hermite Interpolation by Rotation-invariant Spatial Pythagorean-hodograph Curves. Adv. Computational Mathematics 17, 369–383 (2002)
14. Farouki, R.T., Sakkalis, T.: Pythagorean Hodographs. IBM J. Research and Development 34, 736–752 (1990)
15. Kosinka, J., Jüttler, B.: G^1 Hermite Interpolation by Minkowski Pythagorean Hodograph Cubics. Computer Aided Geometric Design 23, 401–418 (2006)
16. Moon, H.P.: Minkowski Pythagorean Hodographs. Computer Aided Geometric Design 16, 739–753 (1999)
17. Peternell, M., Pottmann, H.: A Laguerre Geometric Approach to Rational Offsets. Computer Aided Geometric Design 15, 223–249 (1998)
18. Pottmann, H.: Rational Curves and Surfaces with Rational Offsets. Computer Aided Geometric Design 12, 175–192 (1995)
19. Pottmann, H., Peternell, M.: Applications of Laguerre Geometry in CAGD. Computer Aided Geometric Design 15, 165–186 (1998)
20. Pottmann, H., Wallner, J.: Computational Line Geometry. Springer, Berlin (2001)
21. Ramanathan, M., Gurumoorthy, B.: Constructing Medial Axis Transform of Extruded and Revolved 3D Objects with Free-Form Boundaries. Computer-Aided Design 37, 1370–1387 (2005)
22. Sampl, P.: Medial Axis Construction in Three Dimensions and its Application to Mesh Generation. Eng. Comput. 17, 234–248 (2001)
23. Walrave, J.: Curves and Surfaces in Minkowski Space. Doctoral thesis, K.U. Leuven, Fac. of Science, Leuven (1995)

Mean Value Bézier Surfaces

Torsten Langer and Hans-Peter Seidel

MPI Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{langer, hpseidel}@mpi-inf.mpg.de

Abstract. Bézier surfaces are an important design tool in Computer Aided Design. They are parameterized surfaces where the parameterization can be represented as a homogeneous polynomial in barycentric coordinates. Usually, Wachspress coordinates are used to obtain tensor product Bézier surfaces over rectangular domains. Recently, Floater introduced mean value coordinates as an alternative to Wachspress coordinates. When used to construct Bézier patches, they offer additional control points without raising the polynomial degree. We investigate the potential of mean value coordinates to design mean value Bézier surfaces.

1 Introduction

Bézier surfaces are based on the works of Bézier, de Casteljau, and Forrest [1,2,3] and remain an important tool in Computer Aided Design until today. The most common representations of Bézier surfaces are simplicial Bézier surfaces

$$P(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda) \tag{1}$$

and tensor product Bézier surfaces

$$P(\mathbf{x}) = \sum_{i,j=0}^{n} b_{ij} B_i^n(x) B_j^n(y) \tag{2}$$

where $\lambda := \lambda(\mathbf{x})$ are the barycentric coordinates of $\mathbf{x} := (x, y)$ with respect to a domain triangle while (2) is defined over the domain $[0, 1]^2$, n is the polynomial degree, b_α and b_{ij} are the control points, and B_α^n and B_i^n are the Bernstein polynomials defined by

$$B_\alpha^n(\lambda) = \frac{n!}{\alpha!}\lambda^\alpha, \qquad B_i^n(x) = \binom{n}{i}(1-x)^{n-i}x^i \tag{3}$$

where we use the multi-index notation $\alpha := (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{N}^3$ with $|\alpha| := \sum_i \alpha_i$, $\alpha! := \Pi_i \alpha_i!$, and $\lambda^\alpha := \Pi_i \lambda_i^{\alpha_i}$.

The representations (1) and (2) are even closer related to each other than it may appear at a first glance. If we set

$$\lambda_1^{\mathrm{W}}(\mathbf{x}) := (1-x)(1-y),\ \lambda_2^{\mathrm{W}}(\mathbf{x}) := x(1-y),\ \lambda_3^{\mathrm{W}}(\mathbf{x}) := xy,\ \lambda_4^{\mathrm{W}}(\mathbf{x}) := (1-x)y, \tag{4}$$

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 263–274, 2007.
© Springer-Verlag Berlin Heidelberg 2007

we can express tensor product Bézier surfaces in the form of (1) as has been noted in [4]. Using multi-indices $\alpha := (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \in \mathbb{N}^4$, we obtain

$$P(\mathbf{x}) = \sum_{i,j=0}^{n} b_{ij} B_i^n(x) B_j^n(y) = \sum_{i,j=0}^{n} b_{ij} \binom{n}{i}\binom{n}{j}(1-x)^{n-i}x^i(1-y)^{n-j}y^j \quad (5)$$

$$= \sum_{i,j=0}^{n} b_{ij} \sum_{\substack{\alpha_2+\alpha_3=i \\ \alpha_3+\alpha_4=j \\ |\alpha|=n}} \frac{n!}{\alpha!}\lambda^{\mathrm{W}}(\mathbf{x})^{\alpha} = \sum_{|\alpha|=n} b_{\alpha_2+\alpha_3,\alpha_3+\alpha_4} B_{\alpha}^n(\lambda^{\mathrm{W}}(\mathbf{x})). \quad (6)$$

Here, the equality of (5) and (6) is due to the fact that $B_i^n(x)B_j^n(y)$ and $B_\alpha^n(\lambda^{\mathrm{W}}(\mathbf{x}))$ form both a partition of unity. Therefore, the equality can be deduced by comparing the coefficients of

$$\sum_{i,j=0}^{n} \binom{n}{i}\binom{n}{j}(1-x)^{n-i}x^i(1-y)^{n-j}y^j$$

$$= \sum_{|\alpha|=n} \frac{n!}{\alpha!}\lambda^{\mathrm{W}}(\mathbf{x})^{\alpha} = \sum_{i,j=0}^{n} \left(\sum_{\substack{\alpha_2+\alpha_3=i \\ \alpha_3+\alpha_4=j \\ |\alpha|=n}} \frac{n!}{\alpha!} \right) (1-x)^{n-i}x^i(1-y)^{n-j}y^j. \quad (7)$$

The coordinate functions λ_i^{W} are known as *Wachspress coordinates* [5]. They can be defined not only for the unit square but within all convex polygons. They are also known as generalized barycentric coordinates because they satisfy the following properties for a point $\mathbf{x}$ inside a polygon which is given by boundary vertices $\mathbf{v}_i$

$$\forall i \ \lambda_i(\mathbf{x}) > 0 \qquad \text{positivity}, \quad (8)$$

$$\sum_i \lambda_i(\mathbf{x}) = 1 \qquad \text{partition of unity}, \quad (9)$$

$$\sum_i \lambda_i(\mathbf{x})\mathbf{v}_i = \mathbf{x} \qquad \text{linear precision}, \quad (10)$$

and these properties uniquely determine the classical barycentric coordinates with respect to a triangle.

Recently, Floater suggested a different set of coordinate functions, which he called *mean value coordinates* [6]. We denote these coordinates by λ_i. Using the notation from Fig. 1, they are defined by

$$\lambda_i = \frac{w_i}{\sum_j w_j}, \qquad w_i = \frac{\tan\gamma_{i-1}/2 + \tan\gamma_i/2}{\|\mathbf{x} - \mathbf{v}_i\|} \quad (11)$$

in the interior of a convex polygon, and they have a continuous extension to the boundary [7]. We list some of their properties.

Proposition 1.1 (Hormann & Floater [7]). *Mean value coordinates λ_i have the following properties:*

1. *Positivity inside the polygon (8), partition of unity (9), and linear precision (10).*
2. *Lagrange property:* $\lambda_i(\mathbf{v}_j) = \delta_{ij}$.
3. *Smoothness:* λ_i *is* C^∞ *everywhere except at the vertices* $\mathbf{v}_j$*, where it is only* C^0.
4. *Linear independence: The functions* $(\lambda_i(\mathbf{x}))_{i=1}^4$ *are linearly independent.*
5. *Edge property:* λ_i *is linear along the edges of the unit square.*

Proposition 1.2. *The four mean value coordinate functions with respect to the unit square* $[0,1]^2$ *are symmetric to themselves and to each other:*

$$\lambda_1(x,y) = \lambda_1(y,x), \quad \lambda_3(x,y) = \lambda_3(y,x), \tag{12}$$

$$\lambda_2(x,y) = \lambda_2(1-y,1-x), \quad \lambda_4(x,y) = \lambda_4(1-y,1-x), \tag{13}$$

$$\lambda_1(x,y) = \lambda_2(1-x,y) = \lambda_3(1-x,1-y) = \lambda_4(x,1-y). \tag{14}$$

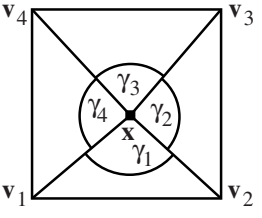


Fig. 1. Notation for the definition of mean value coordinates

Although mean value coordinates are only C^0 at the vertices, we can define partial derivatives by taking the limit over difference quotients within the square. This yields by 1.1.5 $\frac{\partial}{\partial x}\lambda_1(0,0) = \frac{\partial}{\partial x}\lambda_1(1,0) = \frac{\partial}{\partial y}\lambda_1(0,0) = \frac{\partial}{\partial y}\lambda_1(0,1) = -1$, $\frac{\partial}{\partial x}\lambda_1(0,1) = \frac{\partial}{\partial x}\lambda_1(1,1) = \frac{\partial}{\partial y}\lambda_1(1,0) = \frac{\partial}{\partial y}\lambda_1(1,1) = 0$, and respectively for the other coordinate functions. But note that $\lim_{y\to 0, y>0} \frac{\partial}{\partial x}\lambda_1(0,y) \neq -1$ and similar for the other λ_i. Our numerical experiments gave a limit value of approximately -0.7. Nevertheless, this is sufficient for our purposes, and, by taking appropriate combinations of the λ_i, we can achieve higher continuity.

The advantage of mean value coordinates is that their Bernstein polynomials $B_\alpha^n(\lambda)$ are in general different for different αs while the Bernstein polynomials $B_\alpha^n(\lambda^W)$ and $B_{\alpha'}^n(\lambda^W)$ coincide if $\alpha_2+\alpha_3 = \alpha_2'+\alpha_3'$ and $\alpha_3+\alpha_4 = \alpha_3'+\alpha_4'$. This means that it is possible to define a different kind of Bézier patches by using the mean value coordinates of the unit square in (1) as suggested in [6,8]. Moreover, these mean value Bézier patches have a greater number of control points than traditional Bézier patches with the same polynomial degree. In the following section, we will investigate the properties of mean value Bézier surfaces.

2 Mean Value Bézier Surfaces

Many of the properties of (triangular) Bézier surfaces can be proven by formal manipulations of the barycentric coordinates λ_i. Therefore, the respective proofs

carry directly over to the case of mean value Bézier surfaces. In the following theorem, we summarize some of these results.

Proposition 2.1. *Let λ_i be the mean value coordinates, and let the Bernstein polynomials B^n_α and a Bézier patch P be defined as in (3) and (1). Then the following properties hold:*

1. $B^n_\alpha(\lambda) = \sum_{i=1}^4 \lambda_i B^{n-1}_{\alpha-\mathbf{e}_i}(\lambda)$ *(with $B^k_\beta(\lambda) := 0$ if one of the $\beta_i < 0$),*
2. $\{B^n_\alpha\}$ *forms a positive partition of unity inside the unit square,*
3. *the boundary curves $P(\lambda(0,y))$, $P(\lambda(1,y))$, $P(\lambda(x,0))$, and $P(\lambda(x,1))$ are Bézier curves with control points $(b_{i\mathbf{e}_4+(n-i)\mathbf{e}_1})_{i=0}^n$, $(b_{i\mathbf{e}_3+(n-i)\mathbf{e}_2})_{i=0}^n$, $(b_{i\mathbf{e}_2+(n-i)\mathbf{e}_1})_{i=0}^n$, and $(b_{i\mathbf{e}_3+(n-i)\mathbf{e}_4})_{i=0}^n$,*
4. *the image of $[0,1]^2$ under $P(\lambda(\mathbf{x}))$ is contained in the convex hull of the b_α,*
5. *the de Casteljau algorithm works: let $P(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda)$ be a mean value Bézier patch with coefficients b_α. For $k \in \mathbb{N}$ and a given β with $|\beta| = n-k$, let $b^k_\beta(\lambda) := \sum_{|\alpha|=k} b_{\beta+\alpha} B^k_\alpha(\lambda)$. Then $P(\lambda) = b^n_{\mathbf{0}}(\lambda)$ can be computed from the $b^0_\beta(\lambda) = b_\beta$ via the recursive relation $b^k_\beta(\lambda) = \sum_{i=1}^4 \lambda_i b^{k-1}_{\beta+\mathbf{e}_i}(\lambda)$.*

($\mathbf{e}_i$ denotes the multi-index with components $(\mathbf{e}_i)_j = \delta_{ij}$, and $\mathbf{0}$ denotes the multi-index with components $\mathbf{0}_j = 0$.)

Next we give the derivatives of the Bernstein polynomials in λ.

Lemma 2.2. *Let*

$$B^n_\alpha(\lambda) = \begin{cases} \frac{n!}{\alpha!}\lambda^\alpha & \text{if all } \alpha_i \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Then

$$\frac{\partial^{|\beta|}}{\partial\lambda^\beta} B^n_\alpha(\lambda) := \frac{\partial^{|\beta|}}{\partial\lambda_1^{\beta_1}\partial\lambda_2^{\beta_2}\partial\lambda_3^{\beta_3}\partial\lambda_4^{\beta_4}} B^n_\alpha(\lambda) = \frac{n!}{(n-|\beta|)!} B^{n-|\beta|}_{\alpha-\beta}(\lambda) \ . \tag{16}$$

The computation of derivatives of mean value Bézier patches with respect to x and y, which is important to join several Bézier patches smoothly, is more challenging because these derivatives can not be expressed as a linear combination of Bernstein polynomials as it is the case with tensor product Bézier surfaces. Application of the chain rule yields:

Lemma 2.3. *Let*

$$P(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda), \tag{17}$$

and define recursively

$$\Delta_{0,0} b_\alpha(\mathbf{x}) := b_\alpha, \tag{18}$$

$$\Delta_{r+1,s} b_\alpha(\mathbf{x}) := \sum_{i=1}^4 \frac{\partial}{\partial x}\lambda_i(\mathbf{x}) \Delta_{r,s} b_{\alpha+\mathbf{e}_i}(\mathbf{x}), \tag{19}$$

$$\Delta_{r,s+1} b_\alpha(\mathbf{x}) := \sum_{i=1}^4 \frac{\partial}{\partial y}\lambda_i(\mathbf{x}) \Delta_{r,s} b_{\alpha+\mathbf{e}_i}(\mathbf{x}). \tag{20}$$

Then the first derivatives of P are given by

$$\frac{\partial}{\partial x}P(\lambda(\mathbf{x})) = n \sum_{|\alpha|=n-1} \Delta_{1,0} b_\alpha(\mathbf{x}) B_\alpha^{n-1}(\lambda(\mathbf{x})), \tag{21}$$

$$\frac{\partial}{\partial y}P(\lambda(\mathbf{x})) = n \sum_{|\alpha|=n-1} \Delta_{0,1} b_\alpha(\mathbf{x}) B_\alpha^{n-1}(\lambda(\mathbf{x})). \tag{22}$$

The second derivatives of P are given by

$$\frac{\partial^2}{\partial x^2}P(\lambda(\mathbf{x})) = n \sum_{|\alpha|=n-1} \frac{\partial}{\partial x}\Delta_{1,0} b_\alpha(\mathbf{x}) \cdot B_\alpha^{n-1}(\lambda(\mathbf{x})) \tag{23}$$

$$+ n(n-1) \sum_{|\alpha|=n-2} \Delta_{2,0} b_\alpha(\mathbf{x}) B_\alpha^{n-2}(\lambda(\mathbf{x})), \tag{24}$$

$$\frac{\partial^2}{\partial y^2}P(\lambda(\mathbf{x})) = n \sum_{|\alpha|=n-1} \frac{\partial}{\partial y}\Delta_{0,1} b_\alpha(\mathbf{x}) \cdot B_\alpha^{n-1}(\lambda(\mathbf{x})) \tag{25}$$

$$+ n(n-1) \sum_{|\alpha|=n-2} \Delta_{0,2} b_\alpha(\mathbf{x}) B_\alpha^{n-2}(\lambda(\mathbf{x})), \tag{26}$$

$$\frac{\partial^2}{\partial x \partial y}P(\lambda(\mathbf{x})) = n \sum_{|\alpha|=n-1} \frac{\partial}{\partial y}\Delta_{1,0} b_\alpha(\mathbf{x}) \cdot B_\alpha^{n-1}(\lambda(\mathbf{x})) \tag{27}$$

$$+ n(n-1) \sum_{|\alpha|=n-2} \Delta_{1,1} b_\alpha(\mathbf{x}) B_\alpha^{n-2}(\lambda(\mathbf{x})). \tag{28}$$

We can now give continuity conditions for connecting mean value Bézier patches.

Theorem 2.4 (C^0-continuity). *Let*

$$P(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda), \quad P'(\lambda') = \sum_{|\alpha|=n} b'_\alpha B_\alpha^n(\lambda') \tag{29}$$

where $\lambda(\mathbf{x})$ is defined with respect to the square $[0,1]^2$, and $\lambda'(\mathbf{x})$ is defined with respect to the square $[1,2] \times [0,1]$. These patches form a continuous surface iff the control points at the connecting line coincide, that is

$$b_{(n-i)\mathbf{e}_2 + i\mathbf{e}_3} = b'_{(n-i)\mathbf{e}_1 + i\mathbf{e}_4} \quad \forall i = 0 \ldots n. \tag{30}$$

Respective conditions hold to join patches along the other domain boundary edges.

Proof. This is an immediate consequence of Prop. 2.1.3. □

To compare higher order derivatives, we need to know more about the partial derivatives of the mean value coordinates. Instead of trying to give an explicit expression for them, we use the following lemma.

Lemma 2.5. *Let $\lambda_1, \dots \lambda_4$ be the mean value coordinates of the unit square. Then*

$$\frac{\partial}{\partial x}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_2(\mathbf{x}) = \frac{\partial}{\partial x}\lambda_3(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_4(\mathbf{x}) = 0, \tag{31}$$

$$-\left(\frac{\partial}{\partial y}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_2(\mathbf{x})\right) = \frac{\partial}{\partial y}\lambda_3(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_4(\mathbf{x}) = 1, \tag{32}$$

$$-\left(\frac{\partial}{\partial x}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_4(\mathbf{x})\right) = \frac{\partial}{\partial x}\lambda_2(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_3(\mathbf{x}) = 1, \tag{33}$$

$$\frac{\partial}{\partial y}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_4(\mathbf{x}) = \frac{\partial}{\partial y}\lambda_2(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_3(\mathbf{x}) = 0. \tag{34}$$

Proof. The linear precision property (10) of the mean value coordinates implies that linear functions f are correctly interpolated by $\sum_{i=1}^{4} \lambda_i(\mathbf{x}) f(\mathbf{v}_i) = f(\mathbf{x})$ where $\mathbf{v}_i$ are the vertices of the unit square. By choosing $f(\mathbf{x}) := y$, we obtain $\lambda_3(\mathbf{x}) + \lambda_4(\mathbf{x}) = y$. Differentiating by x yields the second equality in (31). Everything else follows completely analogous and by using the partition of unity property (9). □

Theorem 2.6 (C^1-continuity I). *A mean value Bézier patch $P(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda)$ is C^1 everywhere (in particular at the vertices) iff it satisfies the smoothness condition*

$$b_{(n-1)\mathbf{e}_i+\mathbf{e}_{i+2}} = b_{(n-1)\mathbf{e}_i+\mathbf{e}_{i+1}} + b_{(n-1)\mathbf{e}_i+\mathbf{e}_{i-1}} - b_{n\mathbf{e}_i}, \ i = 1 \dots 4 \tag{35}$$

(indices of $\mathbf{e}$ *modulo* 4*).*

Proof. By the symmetry of the mean value coordinates, it is sufficient to prove the claim at one vertex and for one partial derivative. By Lemma 2.3 (compare also (40)), we obtain

$$\lim_{\substack{\mathbf{x}\to(1,1)\\ \mathbf{x}\neq 1}} \frac{\partial}{\partial x} P(\lambda(\mathbf{x})) = n\Bigg((b_{n\mathbf{e}_3} - b_{(n-1)\mathbf{e}_3+\mathbf{e}_4}) \lim_{\substack{\mathbf{x}\to(1,1)\\ \mathbf{x}\neq 1}} \frac{\partial}{\partial x}\lambda_3(\mathbf{x}) + (b_{(n-1)\mathbf{e}_3+\mathbf{e}_2} - b_{(n-1)\mathbf{e}_3+\mathbf{e}_1}) \lim_{\substack{\mathbf{x}\to(1,1)\\ \mathbf{x}\neq 1}} \frac{\partial}{\partial x}\lambda_2(\mathbf{x}) \Bigg) \tag{36}$$

and

$$\frac{\partial}{\partial x} P(\lambda(1,1)) = n(b_{n\mathbf{e}_3} - b_{(n-1)\mathbf{e}_3+\mathbf{e}_4}). \tag{37}$$

We have C^1-continuity if and only if (36) and (37) are equal. Using (33), we obtain the claim. □

Theorem 2.7 (C^1-continuity II). *Let P and P' be given as in (29). Let us denote $b_{ijkl} := b_{i\mathbf{e}_1+j\mathbf{e}_2+k\mathbf{e}_3+l\mathbf{e}_4}$ and correspondingly for b'. Then P and P' form a C^1-continuous surface if (30) and (35) are satisfied and*

$$b_{0,n-i,i,0} - b_{0,n-i,i-1,1} = b'_{n-i,0,1,i-1} - b'_{n-i,0,0,i} \quad \forall i = 1 \dots n \tag{38}$$

and

$$b_{0,n-i,i,0} - b_{1,n-i-1,i,0} = b'_{n-i-1,1,0,i} - b'_{n-i,0,0,i} \quad \forall i = 0 \dots n-1. \tag{39}$$

Respective conditions hold to join patches along the other domain boundary edges.

Proof. The derivatives in y-direction coincide by Prop. 2.1.3. In x-direction, we obtain by Lemma 2.3 and (31)

$$\begin{aligned}
&\frac{\partial}{\partial x} P(\lambda(1,y)) = \\
&\quad n\Bigg(\sum_{i=0}^{n-1} (b_{0,n-1-i,i+1,0} - b_{0,n-1-i,i,1}) \binom{n-1}{i} \left(\frac{\partial}{\partial x}\lambda_3 \cdot \lambda_2^{n-1-i}\lambda_3^i\right)(1,y) \\
&\qquad + (b_{0,n-i,i,0} - b_{1,n-1-i,i,0}) \binom{n-1}{i} \left(\frac{\partial}{\partial x}\lambda_2 \cdot \lambda_2^{n-1-i}\lambda_3^i\right)(1,y)\Bigg)
\end{aligned} \tag{40}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial x} P'(\lambda'(1,y)) = \\
&\quad n\Bigg(\sum_{i=0}^{n-1} (b'_{n-1-i,0,1,i} - b'_{n-1-i,0,0,i+1}) \binom{n-1}{i} \left(\frac{\partial}{\partial x}\lambda'_3 \cdot {\lambda'_1}^{n-1-i}{\lambda'_4}^i\right)(1,y) \\
&\qquad + (b'_{n-1-i,1,0,i} - b'_{n-i,0,0,i}) \binom{n-1}{i} \left(\frac{\partial}{\partial x}\lambda'_2 \cdot {\lambda'_1}^{n-1-i}{\lambda'_4}^i\right)(1,y)\Bigg).
\end{aligned} \tag{41}$$

By the symmetry of the coordinate functions, we obtain $\lambda_2^{n-1-i}(1,y) = {\lambda'_1}^{n-1-i}(1,y)$, $\lambda_3^i(1,y) = {\lambda'_4}^i(1,y)$, and $\frac{\partial}{\partial x}\lambda_3(1,y) = -\frac{\partial}{\partial x}\lambda'_4(1,y) \overset{(31)}{=} \frac{\partial}{\partial x}\lambda'_3(1,y)$. By comparing the coefficients, we see that (38) and (39) are sufficient conditions for C^1-continuity. □

Theorem 2.8 (C^2-continuity I). *Let P and P' be given as in (29). Then P and P' form a C^2-continuous surface (C^1-continuous at the vertices) if (30), (38) and (39) are satisfied, and, for all $i = 0 \dots n-1$,*

$$b_{0,n-1-i,i+1,0} + b_{1,n-1-i,i,0} - b_{0,n-1-i,i,1} - b_{0,n-i,i,0} = 0, \tag{42}$$

and, for all $i = 0 \dots n-2$,

$$b_{2,n-2-i,i,0} + b_{0,n-i,i,0} - 2b_{1,n-1-i,i,0} = b'_{n-2-i,2,0,i} + b'_{n-i,0,0,i} - 2b'_{n-1-i,1,0,i}, \tag{43}$$

$$\begin{aligned}
b_{0,n-1-i,i+1,0} + b_{1,n-2-i,i,1} - b_{1,n-2-i,i+1,0} - b_{0,n-1-i,i,1} \\
= b'_{n-1-i,0,0,i+1} + b'_{n-2-i,1,1,i} - b'_{n-2-i,1,0,i+1} - b'_{n-1-i,0,1,i},
\end{aligned} \tag{44}$$

and

$$\begin{aligned} b_{0,n-2-i,i+2,0} + b_{0,n-2-i,i,2} - 2b_{0,n-2-i,i+1,1} \\ = b'_{n-2-i,0,0,i+2} + b'_{n-2-i,0,2,i} - 2b'_{n-2-i,0,1,i+1}. \end{aligned} \tag{45}$$

Respective conditions hold to join patches along the other domain boundary edges.

Proof. The second derivatives in y-direction coincide by Prop. 2.1.3. In x-direction, we obtain by Lemma 2.3 and by differentiating (31) and (33)

$$\begin{aligned} &\frac{\partial^2}{\partial x^2} P(\lambda(1,y)) = \\ &n\Bigg(\sum_{i=0}^{n-1}(b_{0,n-1-i,i+1,0} - b_{0,n-1-i,i,1} - b_{0,n-i,i,0} + b_{1,n-1-i,i,0}) \\ &\qquad \cdot \binom{n-1}{i}\left(\frac{\partial^2 \lambda_3}{\partial x^2}\lambda_2^{n-1-i}\lambda_3^i\right)\Bigg)(1,y) \\ &+n(n-1)\Bigg(\sum_{i=0}^{n-2}(b_{2,n-2-i,i,0}+b_{0,n-i,i,0}-2b_{1,n-1-i,i,0})\binom{n-2}{i}\left(\left(\frac{\partial \lambda_2}{\partial x}\right)^2\lambda_2^{n-2-i}\lambda_3^i\right) \\ &\qquad + (b_{0,n-1-i,i+1,0} + b_{1,n-2-i,i,1} - b_{1,n-2-i,i+1,0} - b_{0,n-1-i,i,1}) \\ &\qquad\qquad \cdot 2\binom{n-2}{i}\left(\frac{\partial \lambda_2}{\partial x}\frac{\partial \lambda_3}{\partial x}\lambda_2^{n-2-i}\lambda_3^i\right) \\ &\qquad + (b_{0,n-2-i,i+2,0} + b_{0,n-2-i,i,2} - 2b_{0,n-2-i,i+1,1}) \\ &\qquad\qquad \cdot \binom{n-2}{i}\left(\left(\frac{\partial \lambda_3}{\partial x}\right)^2\lambda_2^{n-2-i}\lambda_3^i\right)\Bigg)(1,y) \end{aligned} \tag{46}$$

and a similar term for $\frac{\partial^2}{\partial x^2}P'(\lambda'(1,y))$. The rest follows again by comparing the coefficients of P and P'. With (38), (39), and $\frac{\partial^2}{\partial x^2}\lambda_3(1,y) = \frac{\partial^2}{\partial x^2}\lambda'_4(1,y) \overset{(31)}{=} -\frac{\partial^2}{\partial x^2}\lambda'_3(1,y)$ we deduce (42) from the second line of (46). Note that (35) is a special case of (42) and that an analogon of (42) holds for the control points b'_α as well because of (38) and (39). The other three lines of (46) lead to (43)–(45). When computing the mixed derivatives of P and P', we observe that they already coincide by (38) and (39). □

Theorem 2.9 (C^2-continuity II). *A mean value Bézier patch $P(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda)$ is C^2 everywhere (in particular at the vertices) if it satisfies the smoothness conditions (42) for $i = 0, 1, n-1, n$ and, additionally*

$$b_{2,0,n-2,0} + b_{0,2,n-2,0} - 2b_{1,1,n-2,0} = b_{0,0,n,0} + b_{0,0,n-2,2} - 2b_{0,0,n-1,1} \tag{47}$$

and respective conditions for the other vertices.

Proof. The proof proceeds along the lines of the proof of Thm. 2.6. We use Lemma 2.3 to compute $\frac{\partial^2}{\partial x^2} P(\lambda(1,1))$ and $\lim_{\mathbf{x}\to(1,1),\ \mathbf{x}\neq 1} \frac{\partial^2}{\partial x^2} P(\lambda(\mathbf{x}))$ and obtain (47) by comparing the coefficients. Note that we obtain the condition

$$b_{2,0,n-2,0} + b_{0,2,n-2,0} - 2b_{1,1,n-2,0} = b_{0,1,n-1,0} + b_{1,0,n-2,1} - b_{1,0,n-1,0} - b_{0,1,n-2,1} \tag{48}$$

as well, but this equation is already satisfied by (47) and (42). We get additional constraints from evaluating $\frac{\partial^2}{\partial y^2} P(\lambda(1,1))$ and $\lim_{\mathbf{x}\to(1,1),\ \mathbf{x}\neq 1} \frac{\partial^2}{\partial y^2} P(\lambda(\mathbf{x}))$ and from comparing $\frac{\partial^2}{\partial x \partial y} P(\lambda(1,1))$ with its respective limit values. But again, these constraints can already be deduced from (47) and (42). □

Theorems 2.8 and 2.9 together give sufficient conditions to construct a mean value Bézier surface from patches that is C^2-continuous everywhere.

3 Results and Conclusion

We observe that mean value Bézier surfaces are well-suited for modeling surfaces. Although the advantage of the greater number of control points (compared to tensor product Bézier surfaces of the same degree) is diminished by a greater number of constraints, we obtain nevertheless interesting, new possibilities for

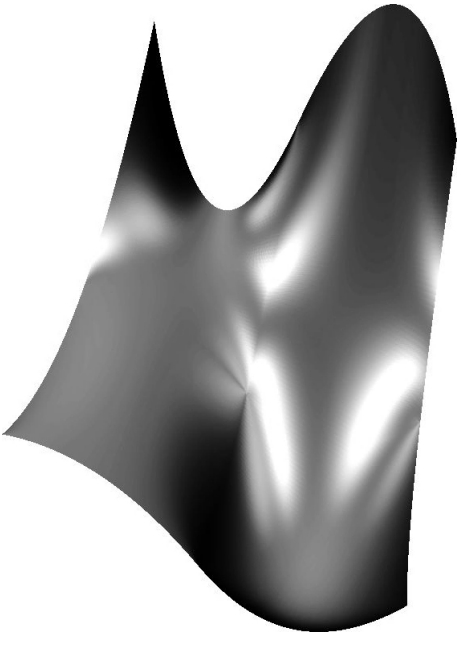


Fig. 2. A mean value Bézier surface constructed from four patches. Here, only the conditions for C^1-continuity along the common boundaries and C^0-continuity at the center vertex were met. This leads to an unpleasant appearance.

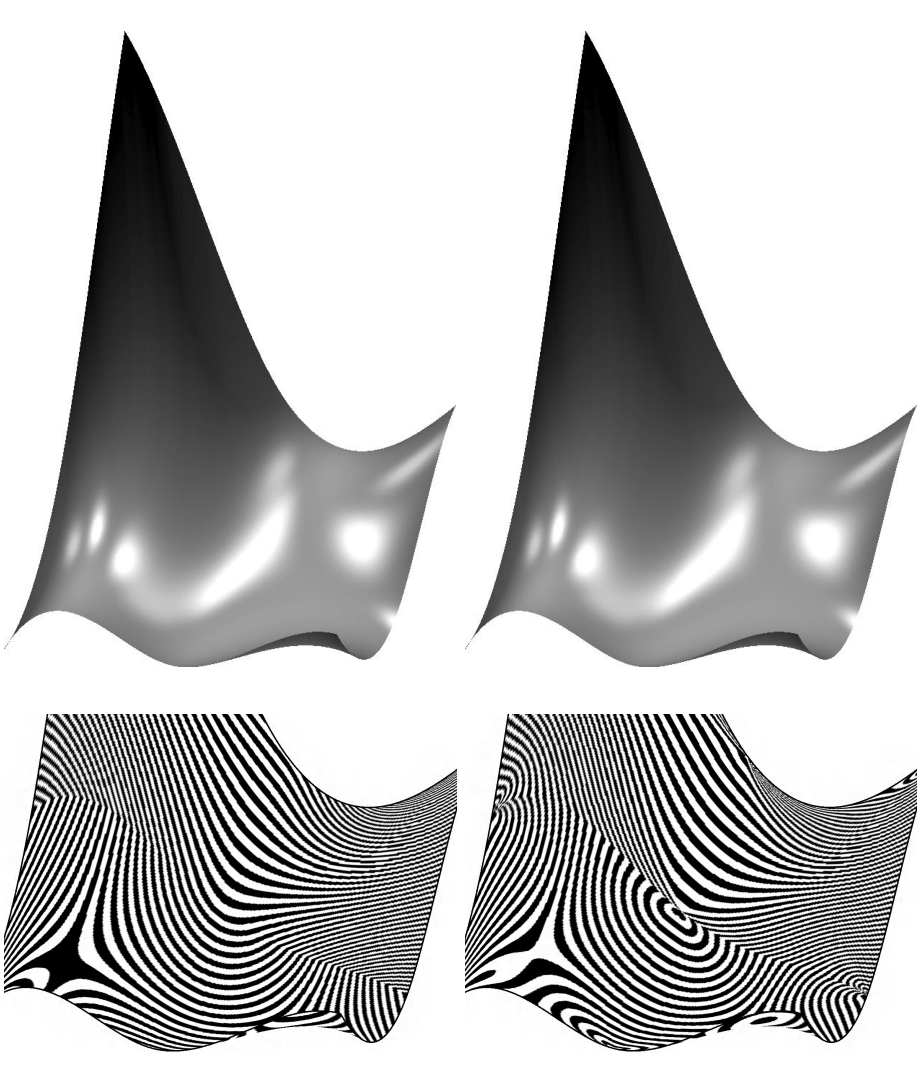


Fig. 3. Comparison of a tensor product Bézier surface (*left*) and a mean value Bézier surface (*right*). Both surfaces are visually indistinguishable which shows the high quality that is achievable with mean value Bézier surfaces. Only the close-up, showing isolines of $\frac{\partial^2}{\partial x^2}P$, reveals the fine differences.

surface construction. One possible, additional benefit is that mean value coordinates, and hence mean value Bézier patches, are well-defined for all convex and non-convex polygons, and we are not restricted to rectangular domains.

In Fig. 2, we show a mean value Bézier surface that is constructed from four patches of order 2. Here, only the conditions for C^1-continuity along the common boundaries were met ((30), (38), and (39)). The central vertex is only C^0. As a result, the joining lines and the central vertex are visible in the shading.

In Fig. 3, we demonstrate that mean value Bézier surfaces and tensor product Bézier surfaces are qualitatively comparable. These surfaces are constructed from four patches of order 3, and the same control points were used in both cases. Here, all constraints to achieve a C^2-continuous surface were enforced, except for the central vertex that is only C^1 for the mean value Bézier surface. This results in two visually indistinguishable surfaces. Only if we compare the isolines of $\frac{\partial^2}{\partial x^2}P$, the C^1-continuous vertex can be discerned.

Mean value coordinates were recently generalized to higher dimensions [9,10,8]. This allows to define mean value Bézier hyper-patches on (hyper-)cubes. A smoothness analysis similar to the one presented in this paper can be done for the resulting hyper-surfaces.

Finally, we want to remark that most of our results are not only valid for constructing Bézier surfaces with mean value coordinates but also for all other differentiable barycentric coordinates as introduced in [11]. This leads to a whole family of different Bézier surfaces. Nevertheless, the convex hull property requires a special property of the mean value (and Wachspress) coordinates: positivity inside the domain. This condition is only guaranteed for few other barycentric coordinates [11] and the reason that we focused on mean value Bézier surfaces.

If Wachspress coordinates are used in our construction of Bézier surfaces, we obtain constant functions for $\frac{\partial}{\partial x}\lambda^{\mathrm{W}}$ and $\frac{\partial}{\partial y}\lambda^{\mathrm{W}}$, and the second derivatives vanish. What remains are the well-known conditions of tensor product Bézier surfaces.

Acknowledgments

We would like to thank Alexander Belyaev and Holger Theisel for helpful discussions. The research of the authors has been supported in part by the EU-Project 'AIM@SHAPE' FP6 IST Network of Excellence 506766.

References

1. Bézier, P.: How Renault uses numerical control for car body design and tooling. Paper SAE 680010, Society of Automotive Engineers Congress, Detroit, Mich (1968)
2. de Casteljau, P.: Outillage méthodes calcul. Paris: André Citroën Automobiles S.A (1959)
3. Forrest, A.R.: Interactive interpolation and approximation by Bézier polynomials. Comput. J. 15(1), 71–79 (1972)
4. Loop, C.T., DeRose, T.D.: A multisided generalization of Bézier surfaces. ACM Transactions on Graphics 8(3), 204–234 (1989)
5. Wachspress, E.L.: A Rational Finite Element Basis. Mathematics in Science and Engineering, vol. 114. Academic Press, New York (1975)
6. Floater, M.S.: Mean value coordinates. Computer Aided Geometric Design 20(1), 19–27 (2003)
7. Hormann, K., Floater, M.S.: Mean value coordinates for arbitrary planar polygons. ACM Transactions on Graphics 25(4), 1424–1441 (2006)

8. Langer, T., Belyaev, A., Seidel, H.P.: Spherical barycentric coordinates. In: Sheffer, A., Polthier, K. (eds.) Fourth Eurographics Symposium on Geometry Processing, Eurographics Association, pp. 81–88 (2006)
9. Floater, M.S., Kós, G., Reimers, M.: Mean value coordinates in 3D. Computer Aided Geometric Design. Special Issue on Geometric Modelling and Differential Geometry 22(7), 623–631 (2005)
10. Ju, T., Schaefer, S., Warren, J.: Mean value coordinates for closed triangular meshes. ACM Transactions on Graphics 24(3), 561–566 (2005)
11. Floater, M.S., Hormann, K., Kós, G.: A general construction of barycentric coordinates over convex polygons. Advances in Computational Mathematics 24(1–4), 311–331 (2006)

Curvature Estimation over Smooth Polygonal Meshes Using the Half Tube Formula

Ronen Lev[1], Emil Saucan[2], and Gershon Elber[3]

[1] OptiTex Ltd., Petach-Tikva 49221, Israel
ronen.lev@optitex.com
[2] Electrical Engineering Department, Technion, Haifa 32000, Israel
semil@ee.technion.ac.il
[3] Computer Science Department, Technion, Haifa 32000, Israel
gershon@cs.technion.ac.il

Abstract. The interest, in recent years, in the geometric processing of polygonal meshes, has spawned a whole range of algorithms to estimate curvature properties over smooth polygonal meshes. Being a discrete approximation of a $\mathcal{C}^2$ continuous surface, these methods attempt to estimate the curvature properties of the original surface. The best known methods are quite effective in estimating the total or Gaussian curvature but less so in estimating the mean curvature.

In this work, we present a scheme to accurately estimate the mean curvature of smooth polygonal meshes using a one sided tube formula for the volume above the surface. In the presented comparison, the proposed scheme yielded results whose accuracy is amongst the highest compared to similar techniques for estimating the mean curvature.

1 Introduction

Polygonal meshes are basic representations of geometry that are employed in a whole variety of fields from vision and image processing to computer graphics, geometric modeling and manufacturing. Analysis of such data sets is of great value in many applications such as reconstruction, segmentation and recognition or even non photorealistic rendering. In this context, curvature analysis plays a major role. As an example, curvature analysis of 3D scanned data sets were shown to be one of the best approaches to segmenting the data [31].

Though many algorithms attempt to estimate the total, or Gaussian curvature K, and mean curvature, H, of the original surface, given its polygonal approximation, we will discuss here only the most well known and accurate ones. For a more rigorous comparison of curvature estimation methods, see, for example, [27].

In [16,25], the principal curvatures and principal directions of a triangulated surface are estimated at each vertex by a least square fitting of an osculating paraboloid to the vertex and its neighbors. These references use quadratic approximation methods where the approximated surface is obtained by solving an over-determined system of linear equations.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 275–289, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In [22], one can find an asymptotic analysis of the paraboloid fitting scheme and an algorithm based on the Gauss-Bonnet Theorem [10], which we refer to as the Gauss-Bonnet scheme, that also known as the *angle deficit* method [22]. In [5,11], the angular deficiency around a vertex is used to estimate the total curvature at a vertex. In a planar domain, the angles around a vertex sums to 2π. The deviation from this value directly reflects on the total curvature at the vertex. A similar estimation is derived for the mean curvature, this time through the dihedral angle of the incoming edges to the vertex.

In [21], circular cross sections, near the examined vertex, are fitted to the surface. Then, the principal curvatures are computed using Meusnier's and Euler's theorems (see [10]). In [28], the principal curvatures are estimated with the aid of an eigenvalues/vectors analysis of 3×3 symmetric matrices that are defined via an integral formulation. In [29], the fact that $\frac{1}{2\pi}\int_0^{2\pi} k_n(\varphi)d\varphi = H$ and $\frac{1}{2\pi}\int_0^{2\pi} k_n(\varphi)^2 d\varphi = \frac{3}{2}H^2 - \frac{1}{2}K$, where $k_n(\varphi)$ is the normal curvature in direction φ, is used to estimate K and H, via discrete approximations of these integrals around a vertex.

In this work, we present a curvature estimation scheme that exploits an infinitesimal volume above the piecewise linear surface to estimate both the total, Gaussian curvature K, and the mean curvature H. It will be shown that this volume above the surface is related to both H as a second order term and to K as a third order term. This identity is then exploited to estimate H accurately and to a lesser extent, K. Further, it will be shown that this formulation is similar to the one offered in [5,11] for the estimation of K and H while more precise results are achieved here for H.

The rest of this paper is organized as follows. In Section 2, we discuss related work and provide the background that is necessary for the proposed algorithm and present the half tube formula. In Section 3, we present the proposed algorithm for the discrete, polygonal case. Then, in Section 4, we provide some examples and compare our results to other curvature estimation algorithms. We summarize our results in Section 5, and give some possible directions for further research.

2 Background and Related Work

Let $S \subset \mathbb{R}^3$ be an orientable surface and let $\bar{N}_p$ denote the unit normal of S at p. For each $p \in S$ consider, in the direction of $\bar{N}_p$, the open symmetric interval of length $2\varepsilon_p$, I_{p,ε_p}, where ε_p is chosen to be small enough such that $I_{p,\varepsilon_p} \cap I_{q,\varepsilon_q} = \emptyset$, for any $p, q \in S$ such that $||p - q|| > \xi \in \mathbb{R}_+$. Then, $Tub_\varepsilon(S) = \bigcup_{p\in S} I_{p,\varepsilon_p}$ is an open set that contains S and such that for any point $x \in Tub_\varepsilon(S)$, there exists a unique normal line to S through x. $Tub_\varepsilon(S)$ is called a *tubular neighborhood* of S or just a *tube*; See Figure 1. The two surfaces $S_{\pm\varepsilon} = S \pm \varepsilon\bar{N}$ are called the *offset surfaces* of S with offset distance ε. We shall consider sets of constant offset ε.

Note that, since ε_p depends upon p, $Tub_\varepsilon(S)$ does not, in general, coincide with the ε-*neigbourhood* of S, i.e., with the set $N_\varepsilon(S) = \{x \in \mathbb{R}^3 \,|\, dist(x, S) = \varepsilon\}$. Also, it is important to know that, while not evident, the existence of tubular

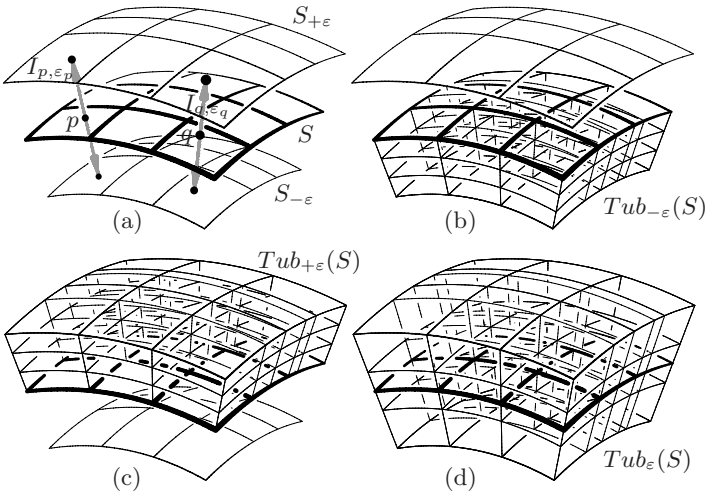


Fig. 1. Tubular neighborhood of S

neighborhoods is assured both locally, for any regular, orientable surface (see [10], Proposition 1, p. 111), and globally for regular, compact, orientable surfaces (see [10], Proposition 3, p. 113).

In addition, we note a number of facts about the existence and regularity of the offset surface S_ε: If S is convex, then $S_{\pm\varepsilon}$ are piecewise $\mathcal{C}^{1,1}$ surfaces (i.e., they admit parameterizations with continuous and bounded derivatives), for all $\varepsilon > 0$. Also, if S is a smooth enough surface with a boundary (that is, at least piecewise $\mathcal{C}^2$), then $S_{\pm\varepsilon}$ are piecewise $\mathcal{C}^2$ surfaces, for all small enough ε (see [12], p. 1025). Moreover, for any compact set $S \subset \mathbb{R}^3$, $S_{\pm\varepsilon}$ are Lipschitz surfaces for almost any ε (see [17]). This is extremely pertinent to our study, since it allows the computation of the mean curvature for local triangulations, not only for smooth surfaces (see also the discussion in Section 5 below).

Theorem 1. (The Tube Formula) *Let $S \subset \mathbb{R}^3$ be a compact orientable surface. Then*

$$\mathrm{Vol}\big(\mathrm{Tub}_\varepsilon(S)\big) = 2\,\varepsilon \mathrm{Area}(S) + \frac{2\varepsilon^3}{3}\int_S K dA\,. \tag{1}$$

Proof. Following [15]. Let $S = f(U)$ be a local parametrization of the surface. Define $f_\delta : U \to \mathbb{R}^3$, $f_\delta(u,v) = f(u,v) + \delta\bar{N}(u,v)$. Then, for a sufficiently small $|\delta|$, f_ε is injective, for all $|\varepsilon| < |\delta|$. Thus, one can choose ε such that $f_{\pm\varepsilon}$ represents a parametrization of $S_{\pm\varepsilon}$. Then:

$$\begin{aligned} \partial f_\varepsilon/\partial u &= \partial f/\partial u + \varepsilon\partial\bar{N}/\partial u, \\ \partial f_\varepsilon/\partial v &= \partial f/\partial v + \varepsilon\partial\bar{N}/\partial v. \end{aligned} \tag{2}$$

However, $II_S(\bar{w}) = -d\bar{N} \cdot \bar{w}$, where II_S denotes the second fundamental form of S (see [10], p. 141). Thus, one can express the partial derivatives of f_ε as:

$$II_S \partial f/\partial u = -\partial \bar{N}/\partial u\,,$$
$$II_S \partial f/\partial v = -\partial \bar{N}/\partial v\,.$$

Therefore (2) becomes:

$$\partial f_\varepsilon/\partial u = (1 - \varepsilon II_S)\partial f/\partial u\,, \tag{3}$$
$$\partial f_\varepsilon/\partial v = (1 - \varepsilon II_S)\partial f/\partial v\,,$$

and

$$\frac{\partial f_\varepsilon}{\partial u} \times \frac{\partial f_\varepsilon}{\partial v} = \det(1 - \varepsilon II_S)\Big(\frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}\Big).$$

Moreover, $\det(1 + \varepsilon I_S) > 0$, for small enough ε, so

$$\left\| \frac{\partial f_\varepsilon}{\partial u} \times \frac{\partial f_\varepsilon}{\partial v} \right\| = \det(1 - \varepsilon II_S) \left\| \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v} \right\|.$$

From the classical formula expressing the principal curvatures of S in terms of H and K (see [10], p. 212, [23], pp. 208-9):

$$\det(1 - \varepsilon II_S) = 1 - 2\varepsilon H + \varepsilon^2 K,$$

it follows (by the well known differential geometry expression for area—see, e.g., [10], 2-8.) that:

$$\begin{aligned}
\mathrm{Area}\big(f_\varepsilon(U)\big) &= \int_U \left\| \frac{\partial f_\varepsilon}{\partial u} \times \frac{\partial f_\varepsilon}{\partial v} \right\| du dv \\
&= \int_U (1 - 2\varepsilon H + \varepsilon^2 K) \left\| \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v} \right\| du dv \\
&= \mathrm{Area}\big(f(U)\big) \;-\; 2\varepsilon \int_{f(U)} H dA \;+\; \varepsilon^2 \int_{f(U)} K dA\,.
\end{aligned}$$

Thus, by summation over the local parameterizations composing S,

$$\mathrm{Area}(S_{\pm\varepsilon}) = \mathrm{Area}(S) \mp 2\varepsilon \int_S H dA + \varepsilon^2 \int_S K dA\,. \tag{4}$$

Therefore, $\mathrm{Area}(S_{+\varepsilon}) + \mathrm{Area}(S_{-\varepsilon}) = 2\mathrm{Area}(S) + 2\varepsilon^2 \int_S K dA$, and it follows that:

$$\mathrm{Vol}\big(\mathrm{Tub}_\varepsilon(S)\big) = \int_{-\varepsilon}^{+\varepsilon} \mathrm{Area}(S_t) dt = 2\varepsilon\, \mathrm{Area}(S) + \frac{2\varepsilon^3}{3} \int_S K dA\,,$$

whence (1) follows immediately. □

By applying the Gauss-Bonnet Theorem for compact surfaces (see [10], p. 276), one immediately gets the following corollary:

Corollary 1. *Let $S \subset \mathbb{R}^3$ be a compact orientable surface. Then:*

$$\mathrm{Vol}\big(\mathrm{Tub}_\varepsilon(S)\big) = 2\,\varepsilon \mathrm{Area}(S) + \frac{4\pi\varepsilon^3}{3}\chi(S)\,, \tag{5}$$

where $\chi(S)$ denotes the Euler characteristic of S.

Evidently, the Tube Formula cannot be employed to compute the mean curvature. Moreover, in the case of triangulated surfaces computing K by means of the Tube Formula reduces to approximating $K(p)$ by the angle defect at the point p (see [8], p. 9 and Remark 1 below); i.e., by approximating $K(p)$ by $K(p) = 2\pi - \sum_1^n \psi_j$, where ψ_j denote the angles of the triangles incident with p (see Figure 4) – that is, by applying a well known method, based upon the Local Gauss-Bonnet Theorem (see [5,20]). Fortunately, (1) also yields a much more useful (yet generally overlooked) formula, which we will refer to as the Half Tube Formula:

Theorem 2. (Half Tube Formula) *Let $S \subset \mathbb{R}^3$ be a compact orientable surface. Then*

$$\mathrm{Vol}\big(\mathrm{Tub}_{\pm\varepsilon}(S)\big) = \varepsilon \mathrm{Area}(S) \mp \varepsilon^2 \int_S H dA + \frac{\varepsilon^3}{3}\int_S K dA\,. \tag{6}$$

Remark 1. A similar formula is developed in [8], p. 4, for *full* tubes, without, however, noticing that the term containing H vanishes. Indeed, our method is closely related to that of [8]. However they stem from somewhat different considerations. Note that, as already mentioned above, only the use of the half tubes formula allows the computation of the mean curvature. Moreover, the method we propose has the advantage of being simpler to implement than the one proposed in [8].

3 The Algorithm

The tube formula is, in fact, a generalization of the classical Steiner-Minkowski Theorem ([1] 12.3.5 , [15] Theorems 10.1 and 10.2.) for compact, convex polyhedra with non-empty interiors of dimension $n \geq 2$, which we present only in the relevant cases, $n = 2$ and $n = 3$:

Theorem 3. (Steiner-Minkowski) *Let $P \subset \mathbb{R}^n$, $n = 2, 3$ be a compact, convex polyhedron and let $N_\varepsilon(P) = \{x \in \mathbb{R}^n \,|\, dist(x, P) \leq \varepsilon\}$, $n = 2, 3$.*

1. *If $n = 2$, then,*

$$\mathrm{Area}(N_\varepsilon(P)) = \mathrm{Area}(P) + \varepsilon \mathrm{Length}(\partial P) + \pi\varepsilon^2\,, \tag{7}$$

where ∂P denotes the boundary of P.

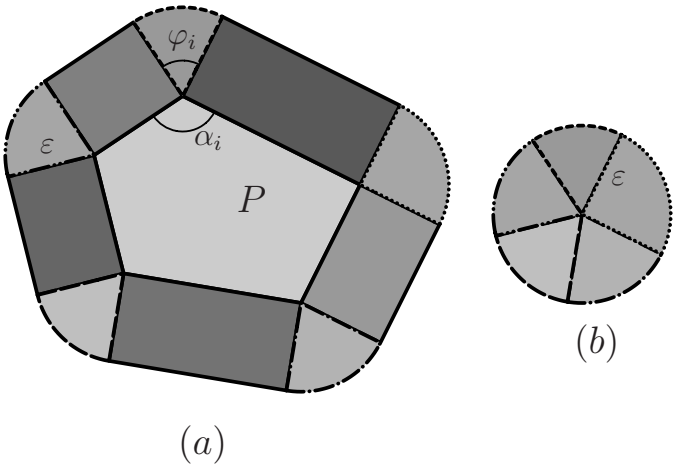


Fig. 2. Tubular neighborhood of a convex polygon

2. If $n = 3$, then,

$$\mathrm{Vol}(N_\varepsilon(P)) = \mathrm{Vol}(P) + \varepsilon \mathrm{Area}(\partial_2 P) + C\varepsilon^2 \mathrm{Length}(\partial_1 P) + \frac{4\pi\varepsilon^3}{3}\,, \qquad (8)$$

where $\partial_2 P$ denotes the boundary faces of P, $\partial_1 P$ denotes the boundary edges of P and the last term contains the 0-dimensional volume contribution of the boundary vertices of P (by convention: $\mathrm{Vol}(\partial_0 P) = |\, V_P \,|$, the number of vertices of P), and where $C = C(P)$ is a scalar value that encapsulates $\iint_S H$ and that essentially depends on the dihedral angles of P.

To gain some insight, we start by analyzing the 2-dimensional case first. $N_\varepsilon(P)$ naturally decomposes into three components: P itself, a union of rectangles of height ε constructed upon the sides of P and the union of circular sectors of radius ε (see Figure 2 (a)). The total area of the rectangles is:

$$\varepsilon \mathrm{Length}(e_1) + \ldots + \varepsilon \mathrm{Length}(e_n) = \varepsilon \mathrm{Length}(\partial P)\,,$$

where $e_1, \ldots, e_n$ denote the boundary edges of P. The angles ϕ_n of the circular sectors are given by: $\phi_i = 2\pi - \pi/2 - \pi/2 - \alpha_i = \pi - \alpha_i$, where α_i is the respective internal angle of P (see Figure 2(a)). However, $\alpha_1 + \ldots + \alpha_n = (n-2)\pi$, thus $\phi_1 + \ldots + \phi_n = 2\pi$. That is, the circular sectors combine to form a disk of radius ε (see Figure 2(b)).

In the case of $n = 3$, $N_\varepsilon(P)$ decomposes into the following four components: P, right parallelipipeds $\mathcal{P}_\varepsilon$ of height ε built upon the faces, the orthogonal products $\mathcal{T}_\varepsilon$ of circular sectors and edges, and spherical sectors $\mathcal{S}_\varepsilon$ associated with the vertices of P and whose union is a ball of radius ε (see Figure 3). Each of the geometrical objects above gives rise to the terms of (8) containing the fitting power of ε.

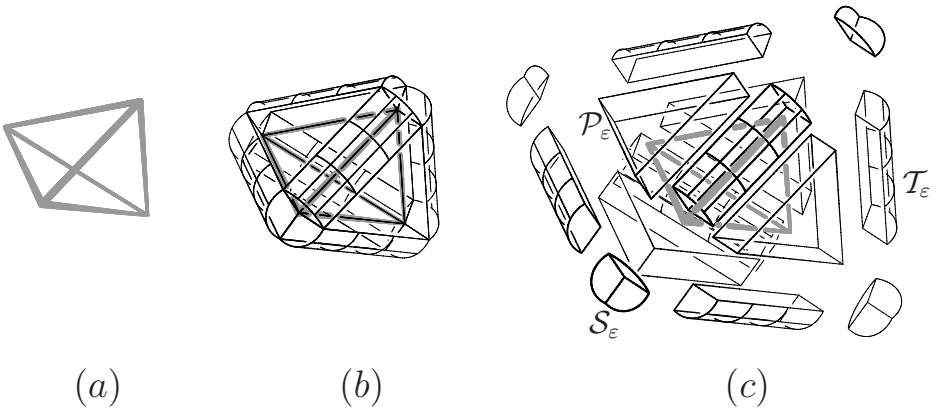


Fig. 3. Tubular neighborhood of convex polyhedron. Note the right parallelepipeds $\mathcal{P}_\varepsilon$ built upon the faces, the orthogonal products $\mathcal{T}_\varepsilon$ of circular sectors and edges, and spherical sectors $\mathcal{S}_\varepsilon$ associated with the vertices.

One immediately notices the similarity between Formulae (4) and (8), more precisely the correspondence between the terms pertaining to H in both of the equations. Beyond its integral expression in (4), due to the limiting process possible on a smooth surface, the main difference resides in the '$\pm$' sign, sign due to discarding the overly restrictive convexity condition.

We denote by $H(p)$ and $K(p)$, the mean and Gauss curvatures at the point p, respectively. Before we proceed further we have to introduce the notion of convergence 'in the sense of measures'. By convergence in the 'sense of measures' we mean the following: given a patch U (i.e., several links of a certain vertex) on the triangulated surface S, each element in U (i.e., vertices, edges, etc.) makes a certain contribution to $H(K)$. Even if these contributions may not individually be correct, they do add to the correct answer *on the average*, when the density of the mesh increases. It is important to note that this averaging effect is not a local phenomenon.

It should be emphasized that while in the computation of $H(p)$, the dihedral angles of the edges through p, $e_1, \ldots, e_n$ (see Figure 4) are computed, this is done only in the sense of measures, this being true for the areas involved in expressing (in the expression of) $K(p)$ (see [8,14]).

That is, one should regard, for instance, Area(T_i) as a weight associated with the triangle T_i and uniformly distributed among its vertices p, q, r. The same uniform distribution is to be considered with respect to the weights naturally associated with edges. Therefore, the measure, i.e., of the dihedral angle associated with the edge e_1, for instance, is to be equally distributed among the vertices adjacent to it, i.e. p and q.

The same type of uniform distribution is to be considered when computing the contribution of Area(T_i), for example in the computation of $K(p)$: T_i will contribute $\frac{1}{3}$Area(T_i) to the computation of (each of) $K(p), K(q), K(r)$ (see Figure 4 [26]).

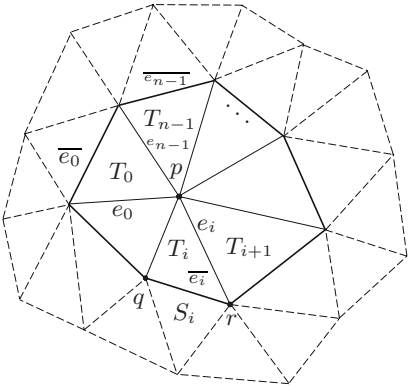


Fig. 4. The first ($\mathrm{Ring}_1(p)$, solid) and second ($\mathrm{Ring}_2(p)$, dashed) ring around vertex p

However, the edge $\overline{e_i} = qr$ also contributes to $H(p)$, since it is an element of T_i, which is adjacent to p. (This may be counterintuitive to the classical approach of computing curvatures of curves *through* p, but one should remember that these 'elementary' edge-curvatures are to be viewed as measures!) Since the boundary edge $\overline{e_i}$ is common to T_i and the second-ring triangle S_i, it's contribution to each of the T_i triangles is half of the associated dihedral angle. Analogous considerations are to be applied in computing the contribution of the boundary vertices (e.g., q), etc.

Let $\mathrm{Ring}_i(p)$ be the i'th ring around p and denote by $|e_i|$ the length of edge e_i. Then, the formula employed for computing the $H(p)$ follows:

$$H(p) = \frac{1}{\mathrm{Area}(\mathrm{Ring}_1(p))}\left[\frac{1}{2}\sum_{i=0}^{n-1}\varphi(T_i, T_{(i+1)\bmod n})|e_i| + \frac{1}{4}\sum_{i=0}^{n-1}\varphi(T_i, S_i)|\overline{e_i}|\right],$$

where $S_i \in \mathrm{Ring}_2(p)$ shares edge $\overline{e_i}$ with $T_i \in \mathrm{Ring}_1(p)$, and $\varphi(T_i, T_{i+1})$ denotes the dihedral angles between adjacent triangles T_i and T_{i+1}.

Algorithm 1 in Figure 5 summarizes this process.

RingArea $\Leftarrow 0$; ▷ The area of the first ring around vertex p
ContribSum $\Leftarrow 0$; ▷ The sum of the contributions of the triangles from the first ring around vertex p
$n \Leftarrow |\mathrm{Ring}_1(p)|$; ▷ Number of triangles in the first ring around vertex p
for $i \leftarrow 0$ to $n-1$ **do**
  RingArea += Area(T_i);
  ContribSum += $\varphi(T_i, T_{(i+1) \bmod n})|e_i| + \frac{1}{2}\varphi(T_i, S_i)|\overline{e_i}|$;
end for
Return ContribSum/(2 RingArea);

Fig. 5. Algorithm 1. Estimate the mean curvature at vertex p.

```
RingArea ⇐ 0;                          ▷ The area of the first ring around vertex p
ContribSum ⇐ 0; ▷ The sum of the contributions of the triangles from the first ring
around vertex p
n ⇐ |Ring_1(p)|;                       ▷ Number of triangles in the first ring around vertex p
for i ← 0 to n − 1 do
    RingArea += Area(T_i);
    ContribSum += T_i.TriangleContrib;
end for
Return ContribSum/(2 RingArea);
```

Fig. 6. Algorithm 2. Estimate the mean curvature at vertex p. Efficient version.

If the user requires the mean curvature for many vertices of the model, this algorithm can be implemented more efficiently as follows: First calculate the contribution of each edge. For each edge, e_i, that is adjacent to triangles T_i and T_{i+1}, we assign an attribute called EdgeContrib that is equal to $\frac{1}{4}\varphi(T_i, T_{i+1})|e|$. Then, for each triangle we assign an attribute called TriangleContrib that is the sum of all EdgeContrib of its edges. Finally, we can compute the mean curvature at vertex p using Algorithm 2 in Figure 6.

4 Tests and Examples

The algorithm described in Section 3 above was tested on a set of synthetic models representing the tessellations of five NURB surfaces: the body and the spout of the infamous 'Utah teapot', a hyperboloid, an ellipsoid and a torus. All these models along with their respective analytic curvature values for each vertex are also available at `http://www.cs.technion.ac.il/~gershon/poly_crvtr/`.

The triangulations of each model were produced for a series of different resolutions, ranging from approximately one hundred triangles to several thousand triangles. Such a distribution of resolutions allowed us gain more insight into the convergence rate of the proposed algorithm and of the various comparison algorithms.

We considered the following mean error value (over all m vertices):

$$\frac{1}{m}\sum_{i=1}^{m}\left|\, H_i - \overline{H}_i \right| ,$$

where H_i denotes the analytically computed value of the mean curvature from the smooth NURBS surface $S(r,t)$ at (r_i, s_i) and $\overline{H}_i$ represents the value of the mean curvature that was estimated by one of the comparison methods at the triangular mesh vertex $v_i = S(r_i, t_i)$.

We compared our algorithm's performance with that of the following previously tested ones: Gauss-Bonnet (or *angle deficit*) [5,11,22], Taubin [28], Watanabe [29] and the classical Parabolic Fit [16,25].

It is evident from the graphs shown in Figures 5–8 that the Tube Formula method gives the best results among the methods (algorithms) for computing

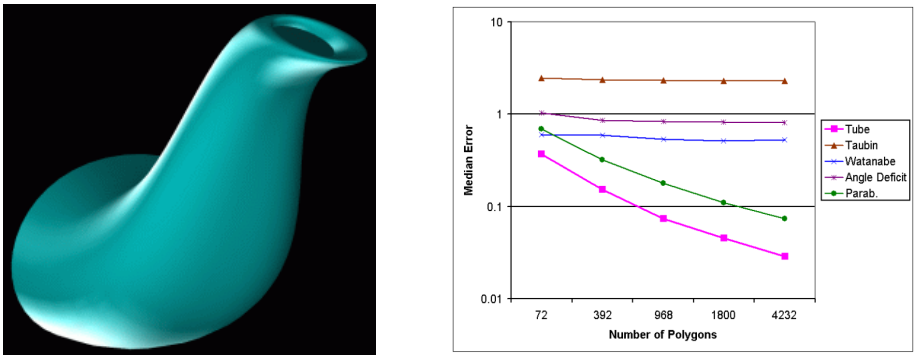


Fig. 7. Average of the absolute error for the value of mean curvature for the tesselations of the Utah teapot's body NURB surface

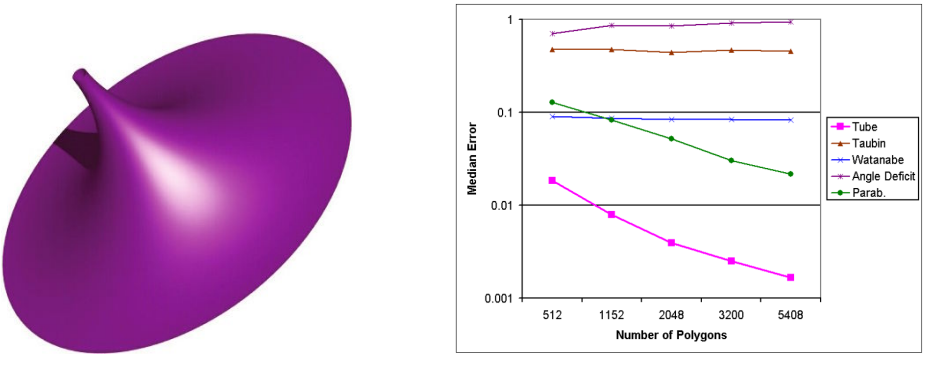


Fig. 8. Average of the absolute error for the value of mean curvature for the tesselations of a rotation surface of negative Gauss curvature

mean curvature, both on surfaces of negative Gauss curvature K (i.e., the hyperboloid) and on surfaces where K takes both positive and negative values (e.g., the spout and the torus). The results obtained when employing our method closely approach those obtained by the best method (e.g., the Parabolic Fit) in the case of surfaces of positive Gauss curvature of high variance (e.g., the ellipsoid).

It fails to produce very good results only for the Body. The probable reason that lays behind this failure is that, in the 'middle section' of the Body, of almost zero Gauss curvature, the tesselation produces patches of (local) negative Gauss curvature, thus introducing the observed error in the computation of the mean curvature. This problem is currently under further investigation.

We conclude this section with a few very brief remarks regarding the computational complexity and time performance of our algorithm: If one denotes (as usual) the number of the vertices of the mesh by $|V|$, then the the algorithm's

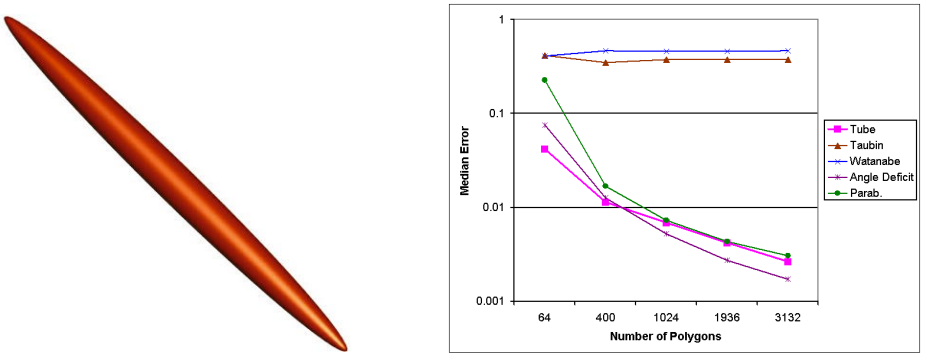


Fig. 9. Average of the absolute error for the value of mean curvature for the tesselations of the ellipsoid

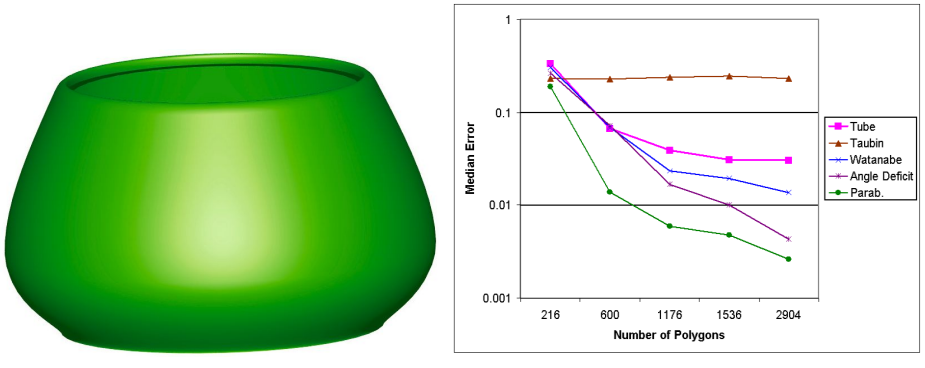
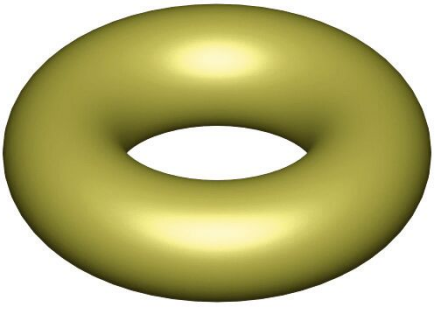


Fig. 10. Average of the absolute error for the value of mean curvature for the tesselations of the Utah teapot's body NURBS surface

complexity is $O(|V|)$. Moreover, as far as the time performance of the algorithm is concerned, the proposed tube formula scheme is demonstrating a comparable speed to the circular fitting method and is about 8 times slower compared to the Taubin, Levin and Watanabe. On a 2 GHz Windows machine, we can process approximately 25k triangles a second.

5 Conclusion and Future Work

As already noted above, it is evident that the Half-Tube Formula method produces, in general, good results, more so for surfaces of negative and mixed Gauss curvature. This is apparently due to the averaging effect of the tubes (or *normal cycles* ([8,9,14]), which represent a generalization of Steiner's approach on convex polyhedra to a much larger class of geometrical objects, in particular to



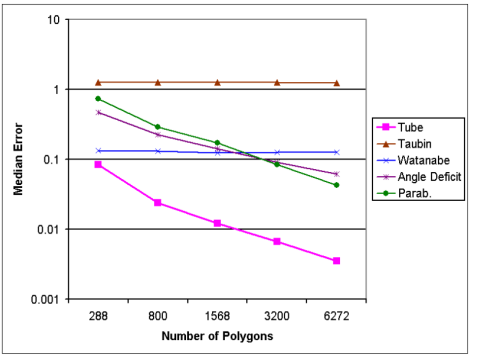


Fig. 11. Average of the absolute error for the value of mean curvature for the tessellations of the torus

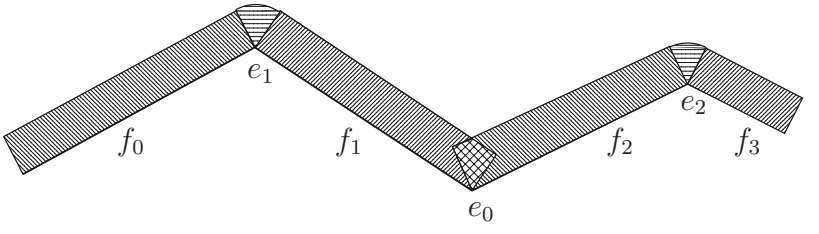


Fig. 12. Normal Cycle at Negative Curvature Vertex

smooth and piecewise-linear manifolds in any dimension) above edges adjacent to vertices of negative curvature (see Figure 10).

Considering the quality results obtained in the case of mean curvature for most types of surfaces, and in light of Formulae (1) and (6), it is natural to next plan to investigate the capability of the Tube and Half-Tube formulas in computing K.

Another direction of study stems from the following facts: Both the Steiner-Minkowski Theorem and the Tube (and Half-Tube) Formula extend not only to arbitrary convex sets in $\mathbb{R}^n$ (see [1]), but also to open subsets with compact closure with smooth boundary in any Riemannian manifold (see [15], pp. 10-11 and 243-248). The most straightforward generalization is the following theorem:

Theorem 4. (Weyl) (See [2], Theorem 6.9.9.) *If X is an k-dimensional submanifold of $\mathbb{R}^n$, then:*

$$\mathrm{Vol}(\mathrm{Tub}_\varepsilon(X)) = \sum_{i=0}^{\lfloor k/2 \rfloor} c_{2i}\varepsilon^{n-k+2i}\,;$$

where

$$c_{2i} = \frac{1}{n-k+2i}\int_X K_{2i}dV_{2i}\,,$$

and where K_{2i} are polynomials of degree i in the curvature tensor R of X, the so-called Weyl curvatures *(see [2], [6]).*

In particular, we have $c_0 = \mathrm{Vol}(X)\mathrm{Vol}(\mathbb{B}^{n-k}(0,\varepsilon))$ and $c_2 = \frac{1}{2}\int_X \kappa\, dX$, where κ denotes the *scalar curvature* (see [2], 6.9, [15], Lemma 4.2). It follows, that in the case of special interest for us, i.e., $n = 3$ and $k = 2$, we have: $\mathrm{Vol}(\mathrm{Tub}_\varepsilon(X)) = c_0\varepsilon + c_2\varepsilon^3$, where $\kappa \equiv K$ – the Gauss curvature of the surface X, i.e., we recover Formula (1).

Moreover, via the normal cycle the results above also extend to piecewise linear manifolds (piecewise linearly and isometrically) embedded in $\mathbb{R}^n$ (see [7,8,13],). Recently these theorems were generalized to general closed sets in $\mathbb{R}^n$ (see [18]).

This last fact is particularly important for the purpose of our study, since it allows us to apply the Half Tube Formula not only locally, that is by computing the measures associated with the triangles belonging to the ring of a given vertex p, but also to extend it to the next ring, and, potentially, to more consecutive rings. Moreover, since the Half Tube Formula involves K also, it allows us estimate K by computing over more than one ring, thus permitting the extension of the angle deficiency method to more than one ring. However, since by (6), the computation of K involves ε^3, one expects difficulties to arise, due to numerical instability.

One further direction of investigation should be to answer the following (rather practical) question: Does the assumption regarding the uniform distribution of the area hold also for 'bad' triangulations (i.e., containing 'thin' triangles), and how much did it affect the quality of the mesh? In view of the seminal results of [6] regarding the good approximations of curvature measures for smooth manifolds by piecewise linear (PL) manifolds whose simplices are 'fat' (i.e., satisfy a certain non-degeneracy condition imposed upon the dihedral angles in all dimensions), one is inclined to conclude that 'fat' triangles should be considered if one wants to make use of the uniform distribution of the area assumption, in which case the original question should be replaced by the more technical one: "What is the lower admissible boundary for the 'fatness' of the triangles if a uniform distribution is to be considered?"

A direct application of our method would be in the computation for of *Willmore (elastic) energy* of triangulated surfaces, where the the Willmore $W(S)$ energy of be a smooth, compact surface S isometrically immersed in $\mathbb{R}^3$ is defined as:

$$W(S) = \int_S H^2 dA\,. \tag{9}$$

Our approach represents an alternative to the one proposed in [3]. Since the Willmore energy is a *conformal invariant* (see [30]), its computation is relevant in a variety of fields and their applications, such as in *conformal geometry*, *variational surface modeling*, *thin structures* (see [4]) and *medical imaging* (see [24]).

Acknowledgment

Emil Saucan was supported by a Viterbi Postdoctoral Fellowship. This work was partially supported by European FP6 NoE grant 506766 (AIM@SHAPE).

References

1. Berger, M.: Geometry II. Universitext. Spinger, Berlin (1987)
2. Berger, M.B., Gostiaux, B.: Differential Geometry: Manifols, Curves and Surfaces. Springer, New York (1987)
3. Bobenko, A.I.: A conformal energy for simplicial surfaces. Combinatorial and Computational Geometry, MSRI Publications 52, 133–143 (2005)
4. Bobenko, A.I., Schröder, P.: Discrete Willmore Flow. In: Desbrun, M., Pottman, H. (eds.) Eurographics Symposium on Geometry Pocessing, pp. 101–110 (2005)
5. Borrelli, V., Cazals, F., Morvan, J.-M.: On the angular defect of triangulations and the pointwise approximation of Curvatures. Computer Aided Geometric Design 20, 319–341 (2003)
6. Cheeger, J., Müller, W., Schrader, R.: On the Curvature of Piecewise Flat Spaces. Comm. Math. Phys. 92, 405–454 (1984)
7. Cheeger, J., Müller, W., Schrader, R.: Kinematic and Tube Formulas for Piecewise Linear Spaces. Indiana Univ. Math. J. 35(4), 737–754 (1986)
8. Cohen-Steiner, D., Morvan, J.M.: Restricted Delaunay triangulations and normal cycle. In: Proc. 19th Annu. ACM Sympos. Comput. Geom. pp. 237–246 (2003)
9. Cohen-Steiner, D., Morvan, J.M.: Approximation of Normal Cycles. Research Report 4723 INRIA (2003)
10. do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall, Englewood Cliffs, N.J. (1976)
11. Dyn, N., Hormann, K., Kim, S.-J., Levin, D.: Optimizing 3d triangulations using discrete curvature analysis. In: Lyche, T., Schumaker, L.L. (eds.) Mathematical Methods in CAGD, Oslo 2000, pp. 135–146 (2001)
12. Federer, H.: Curvature measures. Trans. Amer. Math. Soc. 93, 418–491 (1959)
13. Flaherty, F.J.: Curvature measures for piecewise linear manifolds. Bull. Amer. Math. Soc. 79(1) 100–102
14. Fu, J.H.G.: Convergence of Curvatures in Secant Approximation. J. Differential Geometry 37, 177–190 (1993)
15. Gray, A.: Tubes. Addison-Wesley, Redwood City, CA (1990)
16. Hamann, B.: Curvature approximation for triangulated surfaces. Computing Suppl. 8, 139–153 (1993)
17. Howland, J., Fu, J.H.G.: Tubular neighbourhoods in Euclidean spaces. Duke Math. J. 52(4), 1025–1045 (1985)
18. Hug, D., Last, G., Weil, W.: A local Steiner-type formula for general closed sets and applications. Mathematische Zeitschrift 246(1-2), 237–272 (2004)
19. Kimmel, R., Malladi, R., Sochen, N.: Images as Embedded Maps and Minimal Surfaces: Movies, Color, Texture, and Volumetric Medical Images. International Journal of Computer Vision 39(2), 111–129 (2000)
20. Maltret, J.-L., Daniel, M.: Discrete curvatures and applications, a survey. Preprint (2003)
21. Martin, R.: Estimation of principal curvatures from range data. International Journal of Shape Modeling. 4, 99–111 (1998)

22. Meek, D., Walton, D.: On surface normal and gaussian curvature approximations given data sampled from a smooth surface. Computer Aided Geometric Design 17, 521–543 (2000)
23. O'Neill, B.: Elementary Differential Geometry. Academic Press, N.Y. (1966)
24. Saucan, E., Appleboim, E., Barak, E., Elber, G., Lev, R., Zeevi, Y.Y.: Local versus Global in Quasiconformal Mapping for Medical Imaging. (in preparation)
25. Stokely, E., Wu., S.Y.: Surface parameterization and curvature measurement of arbitrary 3d-objects: Five practicel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(8), 833–840 (1992)
26. Sullivan, J.M.: Curvature Measures for Discrete Surfaces. In: Discrete Differential Geometry: An Applied Introduction. SIGGRAPH2006, pp. 10–13 (2006)
27. Surazhsky, T., Magid, E., Soldea, O., Elber, G., Rivlin, E.: A Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes. In: Proceedings of the IEEE International Conference on Robotics and Automation Taipei, Taiwan 2003, pp. 1021–1026 (2003)
28. Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. ICCV, pp. 902–907 (1995)
29. Watanabe, K., Belyaev, A.G.: Detection of salient curvature features on polygonal surfaces. In: EUROGRAPHICS 2001 Computer Graphics Forum 20(3), 385–392 (2001)
30. Willmore, T.J.: Riemannian Geometry. Clarendon Press, Oxford (1993)
31. Yakoya, N., Levine, M.: Range image segmentation based on differential geometry: A hybrid approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(6), 643–649 (1989)

Segmenting Periodic Reliefs on Triangle Meshes

Shenglan Liu[1,2], Ralph R. Martin[1], Frank C. Langbein[1], and Paul L. Rosin[1]

[1] School of Computer Science, Cardiff University, UK
[2] CMEE, Nanjing University of Aeronautics and Astronautics, China
{Shenglan.Liu, Ralph, F.C.Langbein, Paul.Rosin}@cs.cf.ac.uk

Abstract. Decorative *reliefs* are widely used for e.g. packaging and porcelain design. In *periodic* reliefs, the relief repeats a pattern, for example all the way around an underlying surface of revolution. Reverse-engineering of existing reliefs allows them to be re-applied to different base surfaces; we show here how to segment a single repeat unit of a periodic relief starting from a scanned triangle mesh.

We first briefly review how we segment the relief from the background surface using our previous work. The rest of the paper then concentrates on how we extract a single repeat unit from the relief. To do so, the user provides two points on one relief boundary which are in approximate correspondence on consecutive repeats of the relief. We first refine the relative locations of these points, and then determine a third corresponding point using relief boundary information. These are used to determine three initial cutting planes across the relief. Then surface registration strategies are utilised to refine the correspondence between adjacent repeat units. Finally, we refine the exact locations of the cutting planes by considering only surface information close to the cutting planes. This allows a repeat unit of the periodic relief to be extracted.

We demonstrate that our algorithm is successful and practical, using various real scanned models: user input can be quite imprecise, and we can cope with hand-made reliefs in which the pattern units are only approximately copies of each other.

1 Introduction

Sculptured *reliefs* are widely used in various industries such as sign-making, packaging and ceramics. In many cases, for example when extending or replicating an existing range of products, it is required to reuse the relief designs from existing objects and re-apply them to new CAD models. In many cases, CAD models of the original relief do not exist—for example, the original design for a range of porcelain may be many decades old. Thus, at present it is often necessary for a sculptor to hand-copy existing relief designs, a process that is time consuming, tedious and expensive. An attractive alternative is to reverse engineer the reliefs, allowing automatic extraction and reapplication of previously designed and manufactured reliefs to new CAD models. This approach which would provide benefits to industry in terms of cost savings, and also in time to market.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 290–306, 2007.
© Springer-Verlag Berlin Heidelberg 2007

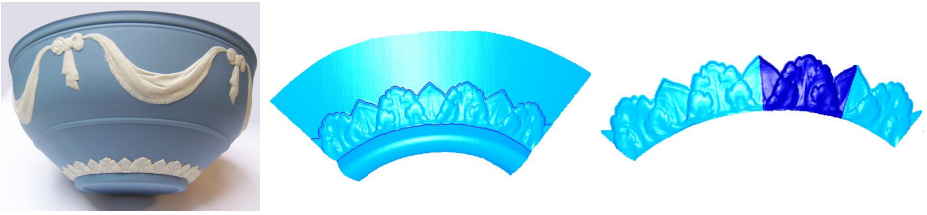


Fig. 1. Periodic relief segmentation from a porcelain bowl: the bowl; segmented partial relief on mesh; an extracted repeat unit

Generally, we may define a relief to be a part of a surface with sculpted features different from the underlying surface, and which is raised by a small height; this height is typically larger than the characteristic size of features on the background. There are various kinds of reliefs, and they can be imposed on diverse backgrounds. The simplest case is that of an isolated relief delimited by a single outer contour, lying on a smooth and slowly varying background. More complex, in terms of processing, are reliefs lying on a textured background. We have already addressed the segmentation problem—separating the relief from the underlying background—for such cases in [1] and [2].

Here, we consider the segmentation of another kind of frequently occurring relief, *periodic reliefs*, illustrated in Fig. 1(left). Here, a basic unit of relief pattern is repeatedly applied to the object. This may be cyclic, forming a closed pattern on objects such as vases, bowls, and bottles for which the underlying surface is a surface of revolution. However, more generally a periodic relief is any relief in which relief units are repeated in a sequence along some arbitrary path (indeed, they could even be repeated two dimensionally to give an overall surface pattern, although we do not consider this case here). In this paper, we give a method designed to extract exactly replicated relief units distributed along, for example, *a circle around a surface of revolution* or *a linear path on a flat surface* to form a closed or open *relief frieze*. Nevertheless, reliefs along *arbitrary paths* may also be extracted using our methods, providing that over a sequence of three successive relief units two criteria are satisfied. Firstly, the shape of the underlying surface should not vary enough to introduce significant changes in the shape of successive relief units—the radii of curvature of the underlying surface should be large compared to the length of a relief unit, and not change much. Secondly, the relief path itself should not be too curved—a large amount of geodesic curvature in the relief path would alter the orientation of the repeat units with respect to the principal directions of the underlying surface, potentially significantly altering the shape of successive relief units. In cases meeting these criteria, successive relief units have sufficiently similar shape that their relationship is reasonably well approximated by a translation and rotation. In fact, many reliefs along more general paths meet these criteria sufficiently well in practice for our methods to work. Furthermore, we note that many existing reliefs of interest are handmade, and so do not exactly repeat anyway, even if e.g. encircling a surface of

revolution. Our methods perform well in the face of such approximate regularity from the outset. Other more complicated cases, in which the relief units are scaled or significantly distorted along the path, are not considered in this paper, however.

To be able to re-apply a periodic relief, our aim in segmentation is to obtain a single repeat unit from the scanned triangle mesh: when the relief is reapplied to a differently-sized or differently-shaped object, a different number of repeats may be required. Thus, there are two subproblems to be considered. Firstly we must segment the relief from the background, using its outer contours. Then we must extract a single repeat unit from the repetitive relief pattern.

The relief's outer boundaries can be detected using our previous segmentation methods: see [1] for the smooth background case, and [2] for textured backgrounds. These methods separate reliefs from their underlying surface using an active contour (or *snake*) driven by energy terms based on two significant relief characteristics, the raised step at the relief boundary, or the difference in surface properties between the relief and the background surface.

Thus, this paper concentrates on the second subproblem, the one of extracting a single relief repeat unit. As noted, if the underlying surface and relief path vary slowly, successive repeat units approximately match each other after applying some rigid transformation. Thus, we use a *surface registration* method to match adjacent repeating units. The repeating units can then be identified by cutting across the relief pattern at corresponding points between adjacent repeats; we do so simply by finding corresponding cutting planes across the relief.

In our final output, the relief generally does not have an exactly identical profile at these cutting planes, for a variety of reasons: because the relief may be hand-made and thus not exactly regular, because of measurement errors, because of errors in registration, and because of minor distortion in the relief units due to the underlying surface shape and relief path. Thus, to make successive repeats a good fit to each other, when placed along a path on a new object, the shape of a repeat unit may need adjustment, particularly in the areas adjacent to the cutting planes. One approach would be to blending the shapes of the left and right-hand ends of the repeat units. Detailed consideration of such adjustment is left as a separate problem for future work; we simply note here that in many ways it is analogous to the well-known problem of 'looping' audio samples.

In Section 2 we review periodicity extraction in related fields. In Sections 3 and 4, we present our algorithm, and discuss details such as the relief boundary period extraction, repeating unit refinement, and cut localization. Results are demonstrated in Section 5. Discussions and conclusions are given in Section 6 and 7.

2 Related Work

The problem of *periodicity* detection and extraction has been studied in the related fields of image processing and computer vision for several decades. *Co-occurrence matrices* have large diagonal entries when sampling corresponds to

the period of a regular repetitive texture, as noted by early researchers [3], and are often used to extract the repetitive cells from images with a regular repetitive texture. For example, Handley [4] used one particular co-occurrence matrix feature, the *dissimilarity*, to characterise the texture periodicity, and then to extract the repeating fundamental unit by finding local maxima.

Many other researchers have adopted another popular periodicity description, the *autocorrelation function*. One such example is provided by [5], where the autocorrelation function is used on an overcomplete wavelet decomposition of an image, together with an efficient peak finding method, to determine the repetitive structure unit in an image texture. Another periodic pattern detection method can be found in [6], where a computational model based on the theory of crystallographic groups is presented. In computer vision, periodic motion detection on image sequences has been studied, such as in [7], where the authors propose a use of time-frequency analysis to a self-similarity measure.

However, the above methods often rely on the regular grid structure of a 2D image, and are thus difficult to directly extend to an irregular 3D triangle mesh. Nevertheless, we find use for the autocorrelation function, by applying it in our algorithm to seeking repetition along the relief boundary curves, after we have resampled points along them at regular intervals. However, it would be much trickier and, and time consuming, to resample the whole mesh into a regular grid.

In computational geometry, relatively little research has focused on periodicity detection in 3D shapes. However, symmetry detection shares certain similarities to our problem, in that both deal with repeating features, and both involve local shape matching under one or a set of transformations. Many different methods have been proposed to find whole-object symmetries with respect to planes through the center of mass. For example, Sun et al. [8] converted the symmetry detection problem to finding correlations in Gaussian images, Martinet et al. [9] introduced a method based on generalised moments, while recently Podolak et al. [10] utilised a planar reflective symmetry transform to capture the reflective symmetries. A method for partial symmetry detection and extraction proposed by Mitra et al. [11] matches simple local shape signatures pairwise to accumulate evidence for symmetries in an appropriate transformation space, then extracts potentially significant symmetries by a clustering method. We note that choosing repeating units in symmetry determination in either the global or partial case does not necessarily require full repeating units to be found. Furthermore, some algorithms work well for exact symmetries, but may not be readily adaptable to approximate symmetries. In our case, we require the repeating units to be determined as completely and accurately as possible within the bounds of their approximate nature, as their boundaries need to be well matched in future reapplication.

3 Algorithm Overview

As mentioned earlier, there are two main steps in our approach for periodic relief segmentation: segmenting the relief from the background surface it lies upon, and

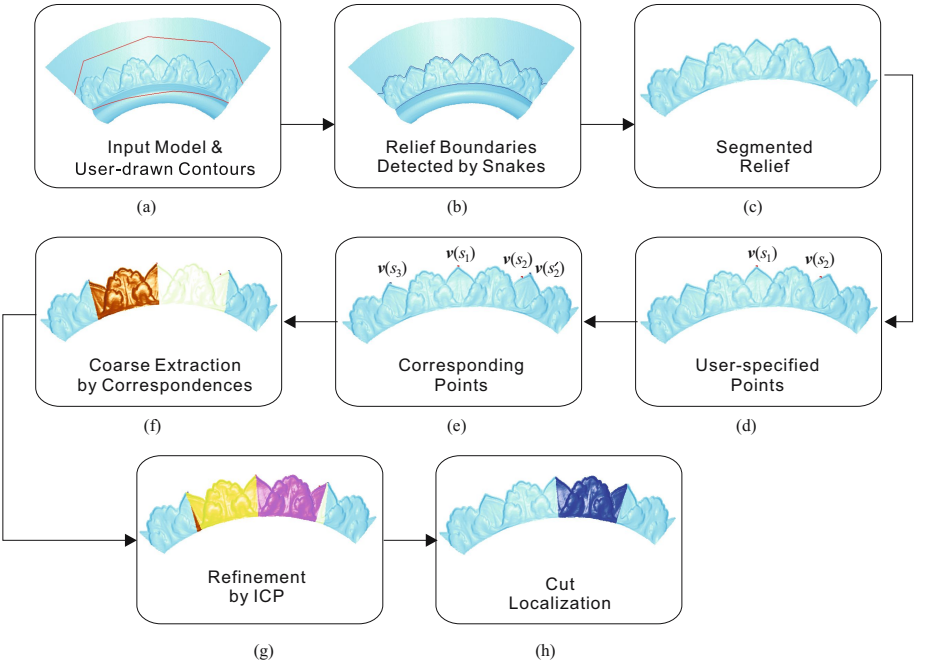


Fig. 2. Periodic relief segmentation—sequence of operations

then extracting a single repeat unit from the relief. The sequence of operations used in our algorithm is illustrated in Fig. 2, where the first three phases belong to the first step, and the remaining phases, the second step.

When segmenting the relief from the background, we can deal with both smooth and textured backgrounds, although in both cases we assume the shape of the background surface varies slowly. In the smooth background case, we use snakes which start from a pair of user-drawn contours, which lie outside the upper and lower boundaries of the relief; their exact locations are unimportant. These evolve until they match the boundary of the relief, determined by using an energy defined to locate each snake at a step feature. If the background is textured, two approaches can be used to segment the relief. The first is to evaluate surface properties at mesh vertices, and use them to define an energy which distinguishes relief from textured background. This is used to drive each snake so that it coarsely matches the relief boundary; it is then finally adjusted to match the desired relief boundary by again seeking a step feature. The second approach is to initially smooth the surface to eliminate the background texture, and allowing the approach used for a smoothed mesh to be applied to find coarse boundary positions; these are again further optimised on the original textured mesh by seeking a step.

Our method for repeat unit extraction starts from two points specified by the user on one relief boundary, which are in approximate correspondence on consecutive repeats of the relief. Three phases are then involved: coarse relief unit

extraction, refinement and cut localization. In the first phase, we first improve the relative locations of the two chosen points on the appropriate boundary curve. We then obtain a new corresponding point on the same boundary curve for a further adjacent relief unit, and from these determine three initial cutting planes across the relief. Next, in the refinement phase, a surface registration strategy based on the *iterative closest point* (ICP) algorithm [12,13] is utilised to find a more accurate match between the coarsely extracted repeating units, allowing us to obtain an optimised unit by refining the locations of the cutting planes. Finally, in the cut localization phase, we update the match between the left- and right-hand edges of the chosen repeat unit using only surface data close to these edges. This produces a better match near the *start* and *finish* of each repeat unit. This is important as in use, copies of the repeat unit are placed next to itself. Details are discussed in the next Section.

4 Algorithm Details

4.1 Whole Relief Segmentation

In this subsection, we remind the reader of certain details of snakes and geometric texture classification from our previous work on relief segmentation. Further information can be found in [1] and [2].

A *snake* is an energy-minimizing spline controlled both by internal forces, such as rigidity and elasticity of the curve which both make it smooth, and tend to shrink, and external forces such as constraint and feature forces which help to drive it towards some desired feature. To deal with the particular problem of relief segmentation, we tailor the snake by careful definition of energy terms, and also carefully control the evolution process. For extraction of reliefs on a smooth background, a feature energy term is specifically designed to detect the step at the edge of a relief. Secondly, a deflation force is used to make the snake move inwards, with strength determined dynamically to balance the internal energy in such a way that the snake is insensitive to choice of initial contour. Thirdly, a refinement phase is used, with different energy terms, to make the snake explore any relief contour concavities.

For reliefs on a textured background, the geometric texture classification method in our algorithm uses a *support vector machine* (SVM) classifier to identify parts of the mesh as belonging to relief or background. The input vectors used for classification are various geometric properties of three kinds: local surface properties, integral properties and statistical properties. The local surface properties are differential geometric quantities such as normals and curvatures, and are estimated using the immediate 1-ring neighbors. The integral properties are computed at multiple scales taken over local neighborhoods [14]. The statistical properties include both first order statistics (such as mean, variance, energy and entropy) [15] and co-occurrence matrix features (such as energy, entropy, contrast and homogeneity) [16], and are extensions of ideas from image texture analysis.

In our earlier work, such techniques were used to extract a relief with a single contour (which could be open or closed). Here, we extract two contours for each relief, which may either be closed (e.g. in the case of a relief encircling a vase), or open (e.g. if we have scanned just part of such a relief).

4.2 Coarse Relief Unit Extraction

It is simpler and faster to use boundary *curves* to extract the initial coarse repeating units than to process pieces of mesh *surface*. In the coarse relief unit extraction phase, the main task is to obtain three points in correspondence on one of the two relief boundaries, and from these, determine three initial cutting planes across the relief delimiting two adjacent, approximately corresponding, pieces of relief. We assume that the user starts by marking two points on the boundary curve which believed to be in approximate correspondence on subsequent relief units. Note that if the user selects arbitrary mesh points, these points generally will not exactly lie on the boundary curves determined previously, so they may have to be moved to the nearest point on the appropriate boundary first. (Alternatively, the user's selection can be constrained to lie on a boundary curve). The user should take into account that the final cutting plane at the left-hand end of the relief unit determined will go through the first of the user's chosen points.

Although periodic reliefs have repetitive boundary curves, in some cases the boundary may be featureless even though the relief itself is not: for example the boundary curve may be a circle, which only trivially shares the period of the relief. We assume here that at least one of the boundary curves has distinct features with the same period as the relief itself, and the user is instructed to select points from that particular boundary. We discuss how to proceed if this assumption is not satisfied later.

To accurately detect the period between corresponding boundary points on a boundary curve, we first resample the particular boundary curve to produce new points at equal arc-length intervals along it. The distance between points is set to the average distance between the original points on the boundary curve. Next, curvature of the boundary curve is estimated at each point, and an autocorrelation of the curvature on the resampled curve is computed. Note that the curvature is independent of a rigid body transformation, making it suitable for determining similarity of parts of the boundary an as-yet unknown transformation. Formally, the autocorrelation of the curvature is defined as follows:

$$A(x) = \frac{\sum_{s=0}^{N} k(s)k(s+x)}{\sum_{s=0}^{N} k^2(s)}, \tag{1}$$

where s is the index of a curve point, x is an integer denoting a candidate periodicity interval, and k denotes the curvature at that point.

The autocorrelation function $A(x)$ has a maximum at $x = 0$ and a number of peaks at other locations corresponding to the curve periodicity. Given the user-chosen point $\mathbf{v}(s_1)$ and its approximate correspondence $\mathbf{v}(s_2)$ on the consecutive

repeat unit, suppose that we believe the approximate user chosen correspondence lies within a distance δ of the true correspondence; we set δ to $(|s_2 - s_1|/2)$ in our algorithm. Then $A(x)$ has a maximum in the range $[s_2 - \delta, s_2 + \delta]$ in Equ. 1. In our program, a simple brute force approach is adequate for seeking the maximum value, since the search range, which is limited to boundary points within two repeat units, is quite small. If we assume that the maximum is at $x = s_p$, then the refined position of the approximate correspondence is $s_2' = s_1 + s_p$. Later, we need three points in correspondence, so we also select another corresponding boundary point $\mathbf{v}(s_3)$ at the peak near $s_1 - s_p$. The large range we use for δ means that once $\mathbf{v}(s_1)$ has been user-selected, the point $\mathbf{v}(s_2)$ need not be specified very carefully at all. For example, in Fig. 2(d), the left red dot is $\mathbf{v}(s_1)$, and the right dot $\mathbf{v}(s_2)$ is quite far from the real correspondence. After improvement using the autocorrelation function, the correspondence is much more accurate as shown by the blue dots in Fig. 2(e).

If the relief boundaries are both featureless, the autocorrelation approach does not work, but we may still proceed provided that the user makes careful estimates of the locations of both initial points. In this case, to find the third point, we simply choose a point at an equal arc-length before $\mathbf{v}(s_1)$ as $\mathbf{v}(s_2)$ is after it. (We discuss the question of why we perform this improvement at all, later).

To estimate the curvature, we follow [17] and use least-squares fitting. In principle, we could use other rigid-body-invariant curve descriptors in place of the curvature, such as the semi-differential invariants proposed by Pajdla and Gool [18,19]. For example, we have tried using the second semi-differential invariant calculated using sliding pairs based on fixed arc-length [19], but found that similar results were produced to those using curvature autocorrelation.

Having found three corresponding points $\mathbf{v}(s_3)$, $\mathbf{v}(s_1)$ and $\mathbf{v}(s_2')$ on one boundary, we now determine three cutting planes at these points. In each case we use the plane which passes through the particular point, and with plane normals, respectively, of $\mathbf{v}(s_3) - \mathbf{v}(s_1)$, $\mathbf{v}(s_3) - \mathbf{v}(s_2')$ and $\mathbf{v}(s_2') - \mathbf{v}(s_1)$. This simple approach gives cutting planes approximately perpendicular to the relief strip, which is generally what the user expects.

As shown in Fig. 2(f), these approximately delimit two repeating units of the relief. Although these planes are not in especially good correspondence, i.e. are not exactly related by the same rotation and translation as the relief itself, they are adequate as a basis for delimiting approximate relief units for the next step.

4.3 Refinement

In this subsection, we refine the coarse relief unit match of the initial repeat units using an ICP algorithm.

The repeating units of a periodic relief are generally only *approximately* congruent, both potentially because of their hand-made nature, and also possibly because of variations in the underlying surface shape, and their orientation upon the underlying surface. We assume that the match is good enough that we can approximate it by a rigid body transformation described as follows:

$$Q = \mathbf{R}P + \mathbf{T}, \tag{2}$$

where P and Q are two point sets belonging to adjacent repeating units (obtained by cutting the relief with the cutting planes), $\mathbf{R}$ is a rotation matrix and $\mathbf{T}$ is a translation vector.

ICP is a well-known algorithm introduced by Besl and McKay [12] for registration and shape matching. It is an iterative alignment algorithm based on minimizing the mean-square distance between two point sets. It works in three phases: 1) establishing correspondence by searching for pairs of nearest points, one from each set, 2) estimating the rigid transformation that overall best maps the first member of each pair onto the second and then 3) applies that transformation to update the position and orientation of the first point set. These three steps are then reapplied until convergence is obtained. The algorithm works quite effectively when given a good initial estimate. Here, we adopt an improved version developed by Eggert et al. [13] which is more robust to errors in initial alignment, and produces a better transformation.

The coarse relief unit extraction phase provides us with three corresponding points on one boundary curve. Thus, as the set of equally-spaced sample points along the curve segment $\mathbf{v}(s_3)\mathbf{v}(s_1)$ match corresponding points on the segment $\mathbf{v}(s_1)\mathbf{v}(s_2')$, these can be used to provide an initial estimate of the rigid body transformation for ICP. Following Besl and McKay [12], we compute this initial transformation using the singular value decomposition (SVD) method.

A refined estimation of the transformation ($\mathbf{R}$ and $\mathbf{T}$) is then produced by the ICP algorithm.

We next apply this transformation and its inverse ($\mathbf{R}^{-1}$ and $-\mathbf{R}^{-1}\mathbf{T}$) to the point $\mathbf{v}(s_1)$, to obtain two new points in *accurate* correspondence with $\mathbf{v}(s_1)$; these replace $\mathbf{v}(s_3)$ and $\mathbf{v}(s_2')$. A refined cutting plane at $\mathbf{v}(s_1)$ is now constructed using the same method as described in Section 4.2, except using these new corresponding points. Finally, two refined cutting planes at the new corresponding points are then computed by applying the found transformation and its inverse to the newly computed plane at $\mathbf{v}(s_1)$. The two repeat units are now more accurately defined by these new cutting planes as shown in Fig. 2(g).

4.4 Cut Localization

In the cut localization phase, we further improve the matching accuracy in regions adjacent to the cutting planes—this is where the relief units join on to each other when being reapplied, and it is more important to get a good match here rather than over the whole of each relief unit: again, note that the repeating units of a periodic relief are generally *only* approximately congruent, for reasons mentioned previously. We perform localization by again using the ICP algorithm, this time on selected areas adjacent to each cut.

The idea is illustrated in Fig. 3. Suppose the two repeating units P and Q are separated by cutting planes as shown at $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. We now optimise the location of the cutting plane at the right-hand end of Q so that Q may be used as a repeat unit. We assume that the cutting plane at the left hand-end of Q remains fixed, to respect the user's selection of the location of v_1 at one end of the repeat unit; furthermore, the normal to this plane is the most reliable of

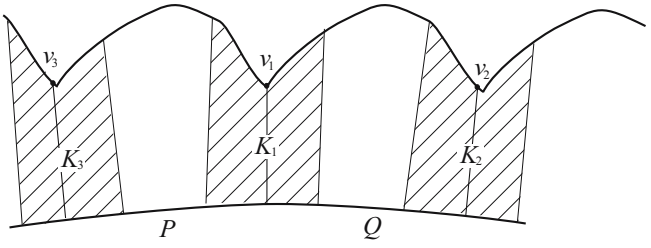


Fig. 3. Key patches

all 3 cutting planes, having been determined by the vector between $\mathbf{v}_2$ and $\mathbf{v}_3$, which is a good approximation to the local direction of the relief strip at $\mathbf{v}_1$. To optimise the location of the right-hand cut on Q, for its use as a repeat unit, it is clear that carefully matching the regions K_1 and K_2 next to the cutting planes will produce a better result than minimizing the matching error between the whole of P and the whole of Q. We call patches K_1 and K_2 *key patches*.

Thus, using as an initialization the transformation found from the refinement phase, we now further optimise the cutting planes by using ICP on the key patches. The width of the key patches can be a fixed width chosen by the user, such as 10% of the repeat unit width. Alternatively, we may iteratively update the cutting planes, reducing the key patch widths at each step. While the latter takes a longer time than using a fixed key patch width, it generally produces better results.

To optimise the repeat unit Q in Fig. 3, we do the following:

1. Start with the whole of the key patch K_1 having half the width of Q (as shown): the cutting plane used to give the left edge of K_1 is found by interpolating by the cutting planes at $\mathbf{v}_3$ and $\mathbf{v}_1$. Similarly, obtain the the cutting plane for the right-hand edge of K_1. Find the left and right cutting planes for K_2 by applying the transformation computed in the refinement phase to the cutting planes for K_1.
2. Register the patches K_1 and K_2 by ICP, giving an updated estimate for the transformation relating the right-hand edge of Q to the left-hand edge.
3. Localize the cutting plane at $\mathbf{v}_2$ by applying the new transformation to the plane at $\mathbf{v}_1$.
4. Decrease the key patch width to half its previous value, centered on the current edges.
5. Repeat steps 1) to 4) until the key patch width is less than a given minimum width, for example, 10% of the repeat unit width.

While the width subdivision method takes a longer time than choosing a fixed key patch width, it also provides us with some important information. If plane movement reduces as we subdivide the key patches, it indicates that a good match has been found between them, giving confidence that the extracted repeat unit is satisfactory. If plane movement increases, this indicates to the user that

another part of the relief might be more suitable for providing a repeat unit. Indeed, the method can readily be automated to proceed to subsequent repeat units as candidates, until a suitable one is found, at the expense of taking longer.

5 Results

Our algorithm has been tested with various real scanned periodic geometric models using a PC with a 2.4GHz CPU and 1GB RAM.

Fig. 4 shows the process of periodic relief segmentation for a model with a quite regular repetitive relief. Fig. 4(a) and (b) show the extraction of the whole relief from the background. Fig. 4(a) shows the scanned model and the initial contours created joining several user-selected points to determine each contour. Fig. 4(b) shows the final positions of the snakes after evolution to the relief boundaries. Figs. 4(c)–(h) illustrate the steps of repeat unit extraction. First, two points on one relief boundary are specified by the user, as shown the red dots in Fig. 4(c). Note that we have deliberately chosen two points far from correct correspondence to show that the algorithm can correct such problems by boundary searching using autocorrelation. This gives a good estimation of the boundary periodicity and refines the correspondence to give the two blue points in Fig. 4(d), which are one period to the right and left of the left-hand red point. The algorithm then coarsely determines two repeat units as shown in Fig. 4(e). Matching the two coarse units using the ICP algorithm refines them as shown in Fig. 4(f). Further ICP matching in the localization phase, using key patches with 20% of the repeat unit width, gives the result shown in Fig. 4(g). Fig. 4(h) shows the final extracted repeat unit, which is the right-hand colored repeat unit in Fig. 4(g). In this model, the repeat units are fairly accurate copies of each other, so the repeat unit after the phases of coarse extraction, refinement and localization are all very similar. In Fig. 4, the original mesh had 102040 triangles and 51451 points, the segmented relief had 53936 triangles and 28036 points, and the repeat unit had approximately 5000 points. The time taken for the whole relief segmentation stage was about 2 minutes. In the relief unit extraction step, the most time consuming parts were uses of the ICP algorithm which took 17 seconds for matching the coarse units in the refinement phase, and another 10 seconds for key patch matching. The other computations took negligible time.

Fig. 5 shows a hand-made periodic leaf relief whose repeat units can, when examined in detail, be seen to be rather irregular in shape. Fig. 5(a) shows the model and the initial contour, and Fig. 5(b) shows the final snake on the relief boundary (there is only one boundary as this relief was at the edge of the mesh). The units separated during the coarse extraction phase, the refinement phase and the localization phase are shown in Figs. 5(c)–(e). It can be seen that the result after localization in Fig. 5(e) is improved compared to the result after refinement in Fig. 5(d). The final extracted unit, the left-hand unit in Fig. 5(e), is shown in Fig. 5(f). Here, we have selected the tip of the leaf to delimit the repeat units because it is readily recognizable to the boundary autocorrelation process.

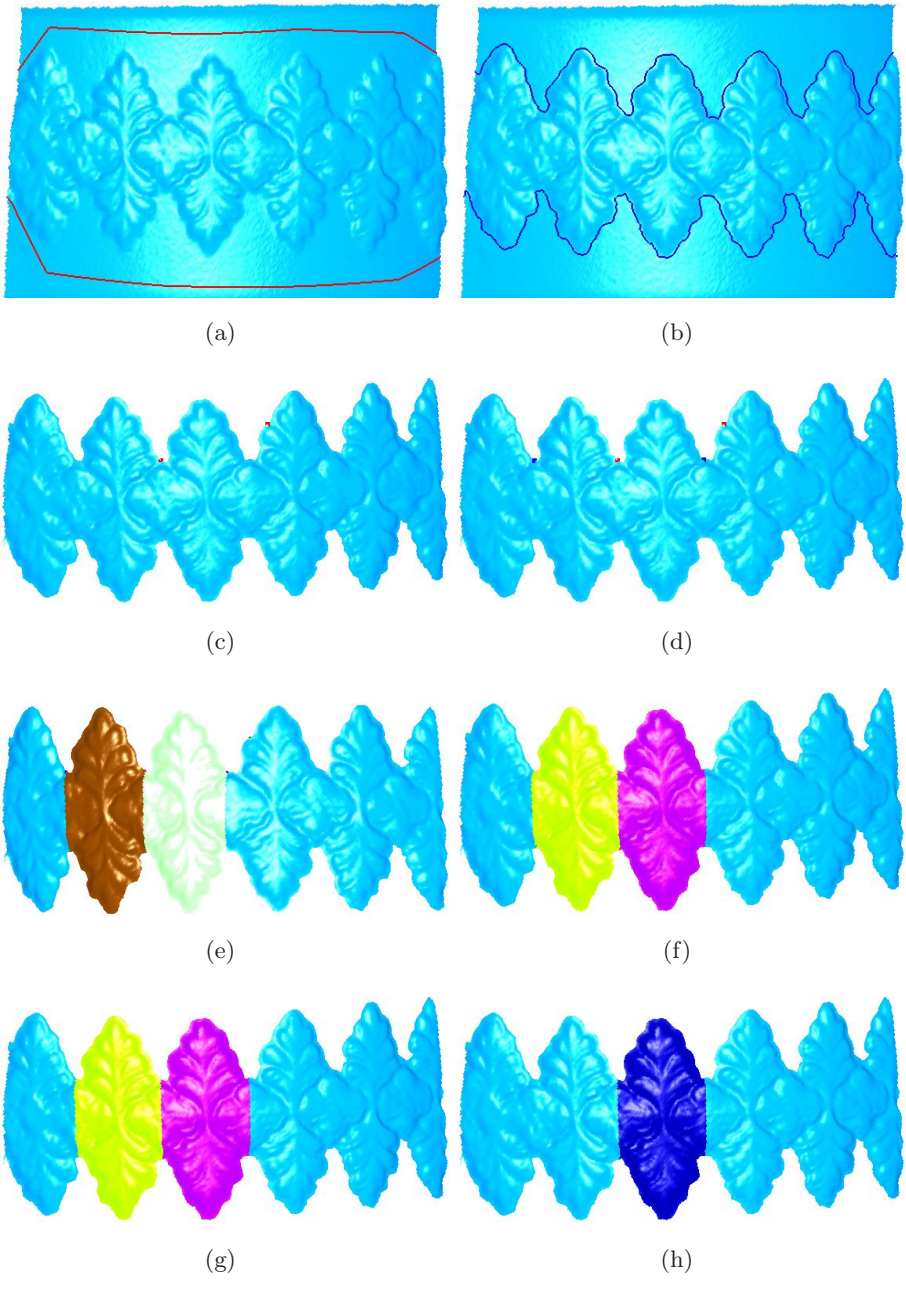


Fig. 4. Periodic relief segmentation for a regular repetitive relief: (a) mesh and initial contour, (b) extracted relief boundaries, (c) two user-specified points in approximate correspondence on consecutive repeats, (d) accurate correspondences, (e) coarse repeat units, (f) refined repeat units, (g) verified refined units, (h) the final extracted unit

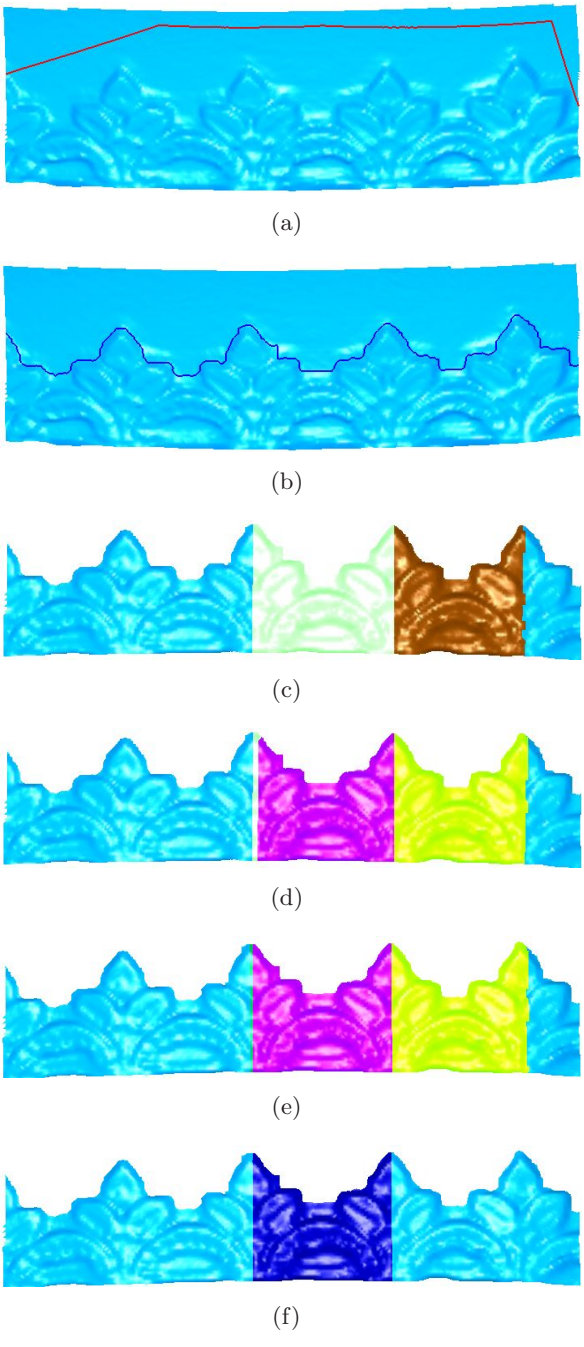


Fig. 5. Periodic relief segmentation for an example with leaves: (a) mesh model and the initial contour, (b) extracted relief boundaries by snake evolution, (c) units separated from coarse extraction phase, (d) results from refinement phase, (e) results from localization phase, (f) final extracted unit

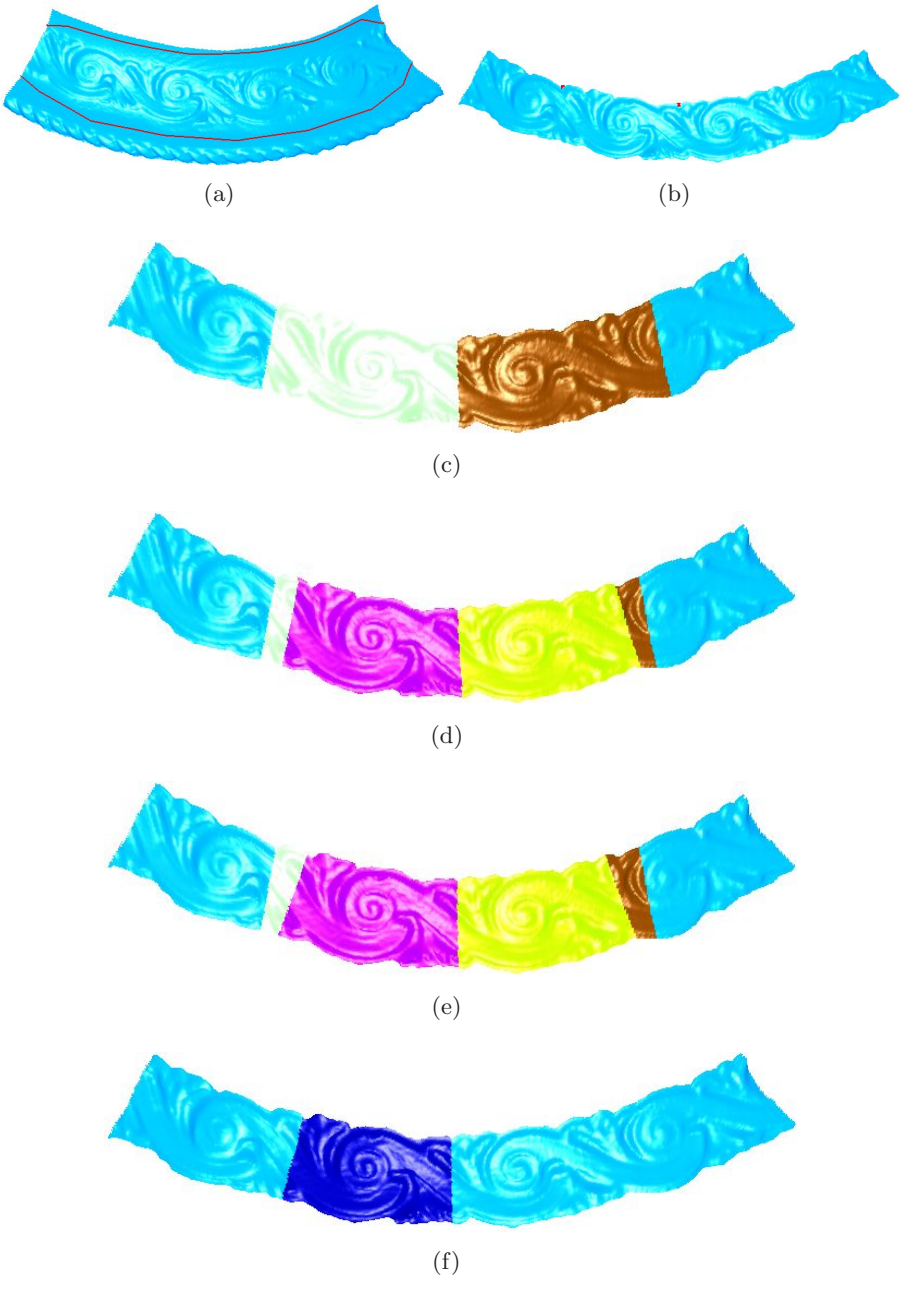


Fig. 6. Periodic relief segmentation without using boundary repetitive extraction: (a) mesh model and the initial contour, (b) user-specified corresponding points, (c) units separated from coarse extraction phase, (d) results from refinement phase, (e) results from localization phase, (f) final extracted unit

Fig. 6 shows an example of periodic relief segmentation in which we do not use boundary autocorrelation. Fig. 6(a) shows the model and the initial contour. Fig. 6(b) shows the two points specified by user. However, the extracted relief boundaries are both rather featureless, and at the same time, very noisy. The autocorrelation function did not work in this case, so this step was omitted. The third point was simply chosen an equal arc-length to the left of the left-hand user chosen point, as the right-hand user chosen point was from it. This gave the coarse repeat units shown in Fig. 6(c). The refinement, localization and final results are shown in Figs. 6(d)–(f). Note that relatively large errors in the coarse patches, due to boundary autocorrelation not being used, are corrected at the refinement stage. In this example, we were unable to obtain a satisfactory result at the localization stage if key patch matching was performed just a single time—even when we tried several different key patch widths. Nevertheless, using the strategy of key patch width subdivision produced a good final result.

6 Discussions

In the coarse extraction phase, we use a boundary repetition detection method to refine the initial user-chosen corresponding points. However, in certain cases the boundary may not be periodic, or if so, relatively featureless, as shown in Fig. 6. Nevertheless, even without accurately corresponding points, good results can still be produced. Why then should we keep the boundary autocorrelation function for improving the initial correspondence? There are mainly two reasons. The first is to provide better coarse units for the downstream ICP algorithm—good initial estimations can lead to quicker ICP convergence and can save a great deal of ICP computing time. The second is that the ICP algorithm is less likely to converge to the wrong local minimum.

In our tests, the ICP algorithm seems to work robustly when the coarse units have at least 70–80% overlap. Thus, if we do not use boundary autocorrelation, the user must specify the correspondence points within a distance of 10–15% of a repeat unit from their accurate locations, which is not particularly difficult for the user.

In some cases, if the coarse stage produces good results, it may be possible to skip the refinement phase. For example, for the model in Fig. 4, if the key patches phase is used immediately after the coarse extraction phase, we obtain similar results.

7 Conclusions and Future Work

Our tests have shown that the proposed algorithm does a good job of extracting relief units. It requires very little user interaction, and is robust to poorly chosen user input. It works on hand-made reliefs in which the repeat units are not exact copies. We conclude by summarizing several important features:

1. A snake-based method is used to evolve a coarsely specified contour to each relief boundary, which gives good whole relief segmentation results.

2. A relief boundary repetition detection method, based on autocorrelation of curve curvature is used to extract coarse repeat units.
3. A surface registration strategy is utilised to refine the match between repeat units.
4. Further localization is done on key patches of the units, near their boundaries, to ensure a good match in the region where adjacent repeat units meet.

Some patterns have local reflective symmetry as well as translational symmetry, e.g. the leaf pattern in Fig. 5. In future, we aim to investigate key matching of reflected units as well, in order to extract the primary semi-pattern.

Acknowledgements

The authors wish to acknowledge the support of Delcam plc, including many helpful discussions with Richard Barratt and Steve Hobbs, and the support of EPSRC grant GR/T24425, for this work.

References

1. Liu, S., Martin, R.R., Langbein, F.C., Rosin, P.L.: Segmenting reliefs on triangle meshes. In: Proceedings ACM Symposiam Solid and Physical Modeling, pp. 7–16. ACM, New York (2006)
2. Liu, S., Martin, R.R., Langbein, F.C., Rosin, P.L.: Segmenting geometric reliefs from textured background surfaces. Computer-Aided Design and Applications (to appear) (2007)
3. Zucker, S., Terzopoulos, D.: Finding structure in co-occurrence matrices for texture analysis. Computer Graphics and Image Processing 2(12), 286–308 (1980)
4. Handley, C.: The analysis and reconstruction of repetitive textures. In: Computer Graphics International, pp. 273–276 (1998)
5. Hsu, J.T., Liu, L.C., Li, C.: Determination of structure component in image texture using wavelet analysis. In: Proceedings of International Conference on Image Processing, pp. 166–169 (2001)
6. Liu, Y., Collins, R.T., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(3), 354–371 (2004)
7. Cutler, R., Davis, L.S.: Robust real-time periodic motion detection, analysis, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 781–796 (2000)
8. Sun, C., Sherrah, J.: 3d symmetry detection using the extended gaussian image. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(2), 164–169 (1997)
9. Martinet, A., Soler, C., Holzschuch, N., Sillion, F.: Accurately detecting symmetries of 3d shapes. Tech. Rep. RR-5692, INRIA (September 2005)
10. Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., Funkhouser, T.: A planar-reflective symmetry transform for 3d shapes. ACM Transactions on Graphics 25(3), 549–559 (2006)

11. Mitra, N.J., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. ACM Transactions on Graphics 25(3), 560–568 (2006)
12. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
13. Eggert, D., Fitzgibbon, A.W., Fisher, R.B.: Simultaneous registration of multiple range views for use in reverse engineering. In: Proceedings of the 13th International Conference on Pattern Recognition, pp. 243–247 (1996)
14. Pottmann, H., Huang, Q.X., Yang, Y.L., Kolpl, S.: Integral invariants for robust geometry processing. Tech. Rep. 146, Geometry Preprint Series, Vienna Univ. of Techn (2005)
15. Materka, A., Strzelecki, M.: Texture analysis methods: A review. Technical Report (1998)
16. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3 3, 610–621 (1973)
17. Lewiner, T., Gomes Jr., J.D., Lopes, H., Craizer, M.: Least squares estimation of curvature and torsion. Technical Report, Department of Mathematics, Pontificia University, France (2005)
18. Pajdla, T., Gool, L.V.: Matching of 3-d curves using semi-differential invariants. In: Proceedings, Fifth International Conference. Computer Vision, pp. 390–395 (1995)
19. Pajdla, T., Gool, L.V.: Efficient matching of space curves. In: Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns, pp. 25–32 (1995)

Estimation of End Curvatures from Planar Point Data

Xinhui Ma and Robert J. Cripps

Geometric Modelling Group, School of Engineering,
Mechanical and Manufacturing, University of Birmingham, Birmingham, B15 2TT,
UK
r.cripps@bham.ac.uk

Abstract. Given a string of discrete planar points, the estimation of principal curvature vectors using circle fitting and Richardson's extrapolation principle has been considered by several authors. However, these methods can not be directly applied to end points, due to symmetry. This article extends these methods to cope with end points. The method is based on the construction of interpolating circles using the first (or last) four data points. Error analysis suggests that the accuracy of curvature estimation using circle fitting is determined by arc-lengths and derivatives of curvature with respect to arc-length. A comparison is made between the proposed four-point method and the well established three-point method.

1 Introduction

Estimation of end principal curvature vectors is essential in curve and surface modelling. These estimates are used in many geometric processes, including composition, subdivision, and simplification [3,5,8]. This paper investigates the estimation of principal curvature vectors from a string of discrete planar points, focusing especially on the end points. It is assumed that these points lie on some planar curve. Estimation of geometric features of strings of points has been considered by several authors. The methods proposed by [1] and [13] use piecewise linear interpolation to estimate the end tangents, from which end normals can be calculated directly, but they did not consider the end curvature estimates. With the expectation that the accuracy may be improved using higher order interpolation, [10] and [14] investigated circle interpolation from 3 points. 3-point formulae based on polynomial interpolation have been proposed [2,8]. Although both methods give end normal and curvature estimates, it is suggested that a circle-based estimate may be better because in the limit the estimates are exact. It is also expected that the accuracy could be improved by using a formula based on 5 rather than 3 points. A 5-point method was introduced by [12] by fitting two circles, but is only suitable for regularly-spaced data points. [15] extended the method to irregularly-spaced data points. However, neither five point method is applicable to end points. A method based on least squares was proposed by [9], but does not provide a sufficiently clear insight into the geometric

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 307–319, 2007.
© Springer-Verlag Berlin Heidelberg 2007

characteristics of the curvature estimation. This paper presents an improved 4-point estimate for the end curvature vector by fitting two circles using a purely geometric approach.

Given a string of $n+1$ ordered and distinct planar points $\mathbf{r}_i, i = 0, \ldots, n; n \geq 3$, that may be irregularly spaced with respect to arc-length, s, the curvature vector $\mathbf{k}(s)$ at the end point $\mathbf{r}_0$ is estimated. The respective curvature vectors $\mathbf{k}_a$ and $\mathbf{k}_b$ at $\mathbf{r}_0$ of the interpolating circles through $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2\}$ and $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_3\}$ are first constructed. An error analysis on $\mathbf{k}_a$ and $\mathbf{k}_b$ is made using purely geometric quantities, and given in terms of the curvature derivatives with respect to, s. The normal vector at $\mathbf{r}_0$ is estimated using Richardson's extrapolation [4] on $\mathbf{k}_a$ and $\mathbf{k}_b$. The curvature estimate is constructed also from $\mathbf{k}_a$ and $\mathbf{k}_b$ using approximate Richardson's extrapolation. The normal and curvature at $\mathbf{r}_n$ can be estimated using a similar approach.

Some basic definitions of the differential geometry of plane curves are reviewed in section 2. Section 3 analyses the error of the 3-point circle-based curvature vector estimate. The improved 4-point estimate of the end point normal vector is derived in section 4 and used in section 5 to determine the improved 4-point estimate for the end point curvature vector. Section 6 investigates the performance of the proposed curvature vector estimates against the established 3-point formula for a range of data. This leads to the conclusion that in most configurations the 4-point formula is more accurate than the 3-point formula.

2 Differential Geometry of Plane Curves

Consider three points on a sufficiently differentiable, arc-length parameterised planar curve $\mathbf{r} = \mathbf{r}(s)$ in the order $\{\mathbf{r}(s), \mathbf{r}(s+\delta s_0), \mathbf{r}(s+\delta s_1\}$, where $0 < \delta s_0 < \delta s_1$. The unit tangent vector $\mathbf{t}(s)$ is defined by

$$\mathbf{r}'(s) = \frac{d\mathbf{r}(s)}{ds} = \lim_{\delta s_0 \to 0} \frac{\mathbf{r}(s+\delta s_0) - \mathbf{r}(s)}{\delta s_0} = \mathbf{t}(s). \tag{1}$$

The osculating circle of a plane curve at $\mathbf{r}(s)$ is the limiting circle through the points as both $\mathbf{r}(s+\delta s_0)$ and $\mathbf{r}(s+\delta s_1) \to \mathbf{r}(s)$ [7], as shown in Fig. 1. The unit principal normal vector $\mathbf{n}(s)$ is defined as the unit vector connecting $\mathbf{r}(s)$ to the centre of the osculating circle and the radius of curvature $\rho(s)$ is defined to be the radius of the osculating circle. When the points are collinear, the radius of curvature is infinite and the unit $\mathbf{n}(s)$ is not uniquely defined. The curvature $\kappa(s) = 1/\rho(s)$, which is zero when the radius of curvature is infinite. Finally, the principal curvature vector $\mathbf{k}(s) = \kappa(s)\mathbf{n}(s)$ is defined by

$$\mathbf{t}'(s) = \mathbf{r}''(s) = \frac{d\mathbf{t}(s)}{ds} = \lim_{\delta s_0 \to 0} \frac{\mathbf{t}(s+\delta s_0) - \mathbf{t}(s)}{\delta s_0} = \mathbf{k}(s). \tag{2}$$

Substituting $\mathbf{k}(s) = \kappa(s)\mathbf{n}(s)$ into Eq.(2) gives

$$\mathbf{r}''(s) = \kappa(s)\mathbf{n}(s). \tag{3}$$

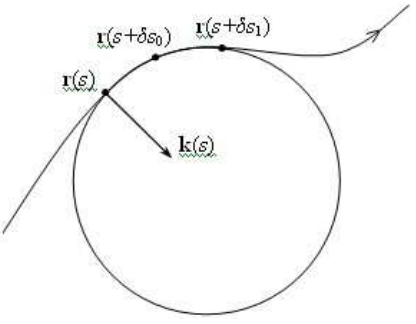


Fig. 1. Osculating circle at $\mathbf{r}(s)$

The signed curvature is defined by the sense of the normal vector. When the sense of the normal is reversed, the sign of the curvature is changed without destroying the governing relation of Eq.(3).

Since $\mathbf{t}(s)$ is a unit vector

$$\|\mathbf{t}(s)\|^2 = \mathbf{t}(s) \cdot \mathbf{t}(s) = 1. \tag{4}$$

Differentiating with respect to s gives

$$\mathbf{t}(s) \cdot \frac{d\mathbf{t}(s)}{ds} = 0, \tag{5}$$

and substituting for $\mathbf{t}'(s)$ from Eq.(2) gives

$$\mathbf{t}(s) \cdot \mathbf{k}(s) = 0, \tag{6}$$

substituting $\mathbf{k}(s) = \kappa(s)\mathbf{n}(s)$ into Eq.(6) gives

$$\mathbf{t}(s) \cdot \kappa(s)\mathbf{n}(s) = 0, \tag{7}$$

i.e., the normal and tangent vectors are perpendicular. The Frenet-Serret formulae for a plane curve are given by [11]

$$\mathbf{t}'(s) = \kappa(s)\mathbf{n}(s), \tag{8}$$

$$\mathbf{n}'(s) = -\kappa(s)\mathbf{t}(s). \tag{9}$$

Differentiating equation Eq.(3) and substituting for $\mathbf{t}'(s)$ and $\mathbf{n}'(s)$ from Eqs.(8,9) gives

$$\mathbf{r}'''(s) = -\kappa^2(s)\mathbf{t}(s) + \kappa'(s)\mathbf{n}(s). \tag{10}$$

Differentiating Eq.(10) gives

$$\mathbf{r}''''(s) = -3\kappa(s)\kappa'(s)\mathbf{t}(s) + (\kappa''(s) - \kappa^3(s))\mathbf{n}(s). \tag{11}$$

Derivatives with respect to s up to 4^{th} order will be used to calculate normal and curvature estimates at the end points in the following sections.

3 Error Analysis of Curvature Vector Estimate

The error in the curvature vector estimated by circle interpolation is given in terms of unit tangent and normal vectors, and hence can be analysed using a geometrically invariant approach. The interpolating circle is constructed through the triplet of points $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2\}$, assumed to be from some underlying smooth curve, and the curvature vector is determined at $\mathbf{r}_0 = \mathbf{r}(s)$. The points are assumed to be possibly irregularly-spaced with respect to arc-length along the curve, i.e.,$\mathbf{r}_0 = \mathbf{r}(s), \mathbf{r}_i = \mathbf{r}(s + s_{i-1}); i = 1, 2, 3$ where $0 < s_0 < s_1 < s_2$. The local geometry and related notation at $\mathbf{r}_0$ are shown in Fig. 2.

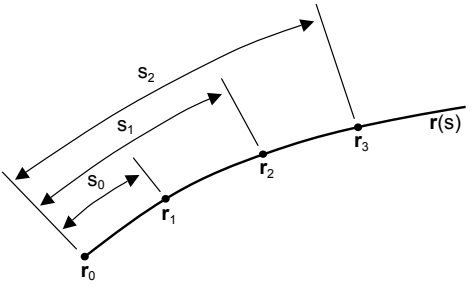


Fig. 2. Local geometry and notation at $\mathbf{r}_0$

The curvature vector $\mathbf{k}_a$ at $\mathbf{r}_0$ of the interpolating circle through points $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2\}$ is given by [6] (see Fig. 3).

$$\mathbf{k}_a = \frac{2\left(\|\mathbf{B}\|^2(\mathbf{A}\cdot\mathbf{C})\mathbf{A} - \|\mathbf{A}\|^2(\mathbf{B}\cdot\mathbf{C})\mathbf{B}\right)}{\|\mathbf{A}\|^2\|\mathbf{B}\|^2\|\mathbf{C}\|^2}, \tag{12}$$

where $\mathbf{A} = (\mathbf{r}_1 - \mathbf{r}_0), \mathbf{B} = (\mathbf{r}_2 - \mathbf{r}_0)$ and $\mathbf{C} = (\mathbf{r}_1 - \mathbf{r}_2)$. Assuming that $\mathbf{r}(s)$ is sufficiently differentiable, it can be expressed as Taylor series expansion. Then the curvature vector $\mathbf{k}_a$ at $\mathbf{r}_0 = \mathbf{r}(s)$ of the interpolating circle through $\{\mathbf{r}(s), \mathbf{r}(s+s_0), \mathbf{r}(s+s_1)\}$ can be constructed as a function of arc-lengths s_0, s_1, and the derivatives of $\mathbf{r}(s)$, and is given by (see Appendix)

$$\begin{aligned}\mathbf{k}_a =& \mathbf{r}''(s) + \frac{(s_0+s_1)}{3}\left((\mathbf{r}''(s)\cdot\mathbf{r}''(s)) + \mathbf{r}'''(s)\right)\\ &+ \frac{(3s_0^2+5s_0s_1+3s_1^2)}{12}\left((\mathbf{r}''(s)\cdot\mathbf{r}'''(s))\mathbf{r}'(s)\right)\\ &+ \frac{(s_0^2+s_0s_1+s_1^2)}{12}\left((\mathbf{r}''(s)\cdot\mathbf{r}''(s))\mathbf{r}''(s) + \mathbf{r}''''(s)\right) + O^3.\end{aligned} \tag{13}$$

In the limit, as s_0 and $s_1 \to 0$, the interpolating circle tends to the osculating circle. This is confirmed by substituting $s_0 = s_1 = 0$ into Eq.(13), i.e. the curvature vector at $\mathbf{r}(s)$ is $\mathbf{r}''(s)$ which is exactly the curvature vector at

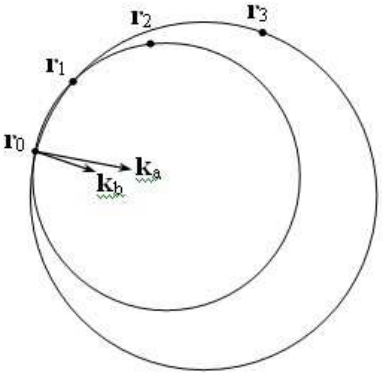


Fig. 3. Interpolating circles for $\mathbf{r}_0$

$\mathbf{r}(s)$ of the osculating circle (*cf* Eq. 3). Now, substituting Eqs.(1,3,10,11) into Eq.(13) gives

$$\begin{aligned}\mathbf{k}_a =& \frac{s_0 s_1}{6}\kappa(s)\kappa'(s)\mathbf{t}(s) \\ &+ \left(\kappa(s) + \frac{(s_0+s_1)}{3}\kappa'(s) + \frac{(s_0^2+s_0s_1+s_1^2)}{12}\kappa''(s)\right)\mathbf{n}(s) + O^3.\end{aligned} \tag{14}$$

This indicates that the accuracy of $\mathbf{k}_a$ is determined by s_0, s_1 and the behaviour of $\kappa(s)$. $\mathbf{k}_a$ has two components; in the tangent and normal directions respectively. Since $\mathbf{k}(s)$ at $\mathbf{r}_0$ is in the normal direction, the tangent component of $\mathbf{k}_a$ is the error component in the tangent direction. The normal component of $\mathbf{k}_a$ contains the curvature $\kappa(s)$ and the error component in the normal direction.

4 Improved Estimate of Normal Vector

The normal vector $\mathbf{n}(s)$ at the end point $\mathbf{r}_0$ is estimated by applying Richardson's extrapolation principle to two estimated curvature vectors $\mathbf{k}_a$ and $\mathbf{k}_b$. Eq.(14) can be rewritten as

$$\mathbf{k}_a = s_0 s_1 F_t \mathbf{t}(s) + F_{na}\mathbf{n}(s) + O^3, \tag{15}$$

where $F_t = \dfrac{\kappa(s)\kappa'(s)}{6}$ and $F_{na} = \kappa(s) + \dfrac{(s_0+s_1)}{3}\kappa'(s) + \dfrac{(s_0^2+s_0s_1+s_1^2)}{12}\kappa''(s)$.

By taking another triplet of points $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_3\}$, the curvature vector $\mathbf{k}_b$ at $\mathbf{r}_0$ of the interpolating circle can be constructed. Richardson's extrapolation principle can be applied to construct an estimate for the normal vector $\mathbf{n}(s)$ at $\mathbf{r}_0$ as a linear combination of $\mathbf{k}_a$ and $\mathbf{k}_b$. Then the curvature vector at $\mathbf{r}_0$ of the interpolating circle through $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_3\}$ can be expressed as

$$\mathbf{k}_b = s_0 s_2 F_t \mathbf{t}(s) + F_{nb}\mathbf{n}(s) + O^3, \tag{16}$$

$$\text{where} \quad F_{nb} = \kappa(s) + \frac{(s_0 + s_2)}{3}\kappa'(s) + \frac{(s_0^2 + s_0 s_2 + s_2^2)}{12}\kappa''(s).$$

Applying Richardson's extrapolation principle to Eqs.(15,16) gives

$$s_2\mathbf{k}_a - s_1\mathbf{k}_b = (s_2 F_{na} - s_1 F_{nb})\,\mathbf{n}(s) + O^3. \tag{17}$$

Comparing the vector directions on each side of Eq.(17) gives

$$\mathbf{n}(s) \approx \frac{s_2\mathbf{k}_a - s_1\mathbf{k}_b}{\|s_2\mathbf{k}_a - s_1\mathbf{k}_b\|}. \tag{18}$$

By eliminate the second order error component in the tangent direction, i.e., $s_0 s_i F_t \mathbf{t}(s)$, an improved estimate for the normal vector at the end points of irregularly-spaced planar points has been constructed. The estimate is invariant with respect to a constant scaling of the arc-lengths and produces the exact result when the data points are taken from a line or circle. The limiting estimate as $s_0, s_1 \to 0$ approach zero is also exact.

5 Improved Estimate of Curvature Vector

Estimates for the curvature vector $\mathbf{k}_a$ at $\mathbf{r}_0$ of the interpolating circle through $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2\}$ (Section 3) and normal vector $\mathbf{n}(s)$ (Section 4) are used to estimate the associated curvature using an improved four-point formula. Then, the curvature vector $\mathbf{k}(s)$ at $\mathbf{r}_0$ is estimated as the product of $\mathbf{n}(s)$ and $\kappa(s)$. Since the accuracy of curvature estimation using circle fitting is determined by arc-lengths and the behaviour of the curvature, the estimate should be a function of s_0, s_1 and derivatives of $\kappa(s)$. The 2^{nd} order derivative of $\kappa(s)$ corresponds to the 4^{th} order derivative of curve $\mathbf{r}(s)$, hence it is assumed that the higher-order derivatives of $\kappa(s)$ can be ignored and only $\kappa'(s)$ and $\kappa''(s)$ are used. Since $\mathbf{t}(s) \cdot \mathbf{n}(s) = 0$ and $\mathbf{n}(s) \cdot \mathbf{n}(s) = 1$ Eq.(14) can be written

$$\mathbf{k}_a \cdot \mathbf{n}(s) = \kappa(s) + \frac{(s_0 + s_1)}{3}\kappa'(s) + \frac{(s_0^2 + s_0 s_1 + s_1^2)}{12}\kappa''(s) + O^3, \tag{19}$$

i.e. $\mathbf{k}_a \cdot \mathbf{n}(s)$ is the exact curvature $\kappa(s)$ plus an error series in which each term is the product of a function of (s_0, s_1) and a derivative of $\kappa(s)$. By taking another triplet of points $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_3\}$, the curvature vector $\mathbf{k}_b$ at $\mathbf{r}(0)$ of the interpolating circle can be constructed. Similarly to Eq.(19), the component of $\mathbf{k}_b$ in the direction of $\mathbf{n}(s)$ can be expressed as

$$\mathbf{k}_b \cdot \mathbf{n}(s) = \kappa(s) + \frac{(s_0 + s_2)}{3}\kappa'(s) + \frac{(s_0^2 + s_0 s_2 + s_2^2)}{12}\kappa''(s) + O^3. \tag{20}$$

In order to improve the curvature estimate by reducing both 1^{st} and 2^{nd} order errors, an approach similar to Richardson's extrapolation can be applied to Eqs.(19,20). $\mathbf{k}_b$ and $\mathbf{k}_b$ are weighted so that they have approximately the same

absolute errors. To determine the weights, assume that the points are regularly spaced, i.e., $s_1 = 2s_0$ and $s_2 = 3s_0$. Substitution into Eqs.(19,20) gives

$$\mathbf{k}_a \cdot \mathbf{n}(s) = \kappa(s) + s_0\kappa'(s) + \frac{7}{12}s_0^2\kappa''(s) + O^3. \tag{21}$$

and

$$\mathbf{k}_b \cdot \mathbf{n}(s) = \kappa(s) + \frac{4}{3}s_0\kappa'(s) + \frac{13}{12}s_0^2\kappa''(s) + O^3. \tag{22}$$

The same 1^{st} order errors can be obtained by applying the weights $(4:3)$ to $\mathbf{k}_a \cdot \mathbf{n}(s)$ and $\mathbf{k}_b \cdot \mathbf{n}(s)$ respectively. Similarly, the same 2^{nd} order errors can be obtained using the weights $(13:7 \approx 2:1)$. Averaging gives balanced weights: $(3:2)$. Since $(s_1 : s_2 = 3:2)$, using $s_2\mathbf{k}_a \cdot \mathbf{n}(s)$ and $s_1\mathbf{k}_b \cdot \mathbf{n}(s)$ will partially eliminate the errors of the curvature estimates. Substituting these weights in Eqs.(21,22) gives

$$(s_2\mathbf{k}_a - s_1\mathbf{k}_b) \cdot \mathbf{n}(s) = (s_2 - s_1)\left(\kappa(s) + \frac{s_0}{3}\kappa'(s) + \frac{(s_0^2 - s_1s_2)}{12}\kappa''(s)\right) + O^3. \tag{23}$$

Dividing by $(s_2 - s_1)$, gives

$$\frac{(s_2\mathbf{k}_a - s_1\mathbf{k}_b) \cdot \mathbf{n}(s)}{s_2 - s_1} = \kappa(s) + \frac{s_0}{3}\kappa'(s) + \frac{(s_0^2 - s_1s_2)}{12}\kappa''(s) + O^3. \tag{24}$$

Substituting $s_1 = 2s_0$ and $s_2 = 3s_0$ into the right side of Eq.24 gives

$$\frac{s_2\mathbf{k}_a - s_1\mathbf{k}_b}{s_2 - s_1} \cdot \mathbf{n}(s) = \kappa(s) + \frac{1}{3}s_0\kappa'(s) - \frac{5}{12}s_0^2\kappa''(s) + O^3. \tag{25}$$

Comparing the error terms of Eq.(25) to those of Eq.(19) illustrates that the estimate can reduce the first and second order errors by $2/3$ and $2/7$ respectively. Therefore, for approximately regularly spaced data points, the analysis suggests

$$\kappa(s) \approx \frac{s_2\mathbf{k}_a - s_1\mathbf{k}_b}{s_2 - s_1} \cdot \mathbf{n}(s) \approx \frac{\|s_2\mathbf{k}_a - s_1\mathbf{k}_b\|}{s_2 - s_1}. \tag{26}$$

Since $\mathbf{k}(s) = k(s)\mathbf{n}(s)$, substituting from Eqs.(18,26), the estimated curvature vector at the end point $\mathbf{r}_0$ is given by

$$\mathbf{k}(s) \approx \frac{s_2\mathbf{k}_a - s_1\mathbf{k}_b}{s_2 - s_1}. \tag{27}$$

The curvature vector at the last point $\mathbf{r}_n$ can also be estimated using a similar approach. Although the estimates will be most appropriate for regularly-spaced points whose high order derivatives of $\kappa(s)$ can be ignored, they can still be used for any point spacing and any curvature behaviour since they has several desirable properties including; well defined and invariant with respect to constant scaling of arc-lengths, exact results for linear and circular data, the limiting estimate as $s_2 \to 0$ is also exact, and the estimate tends towards $\mathbf{k}_a$ in a continuous manner when s_1 decreases, or when s_2 increases.

6 Numerical Validation

In this section, the quality of the 4-point estimates are assessed and compared against the circle-based 3-point method. A range of smooth parametric curves (no cusps or crossings) with known smooth curvature profiles are used in the

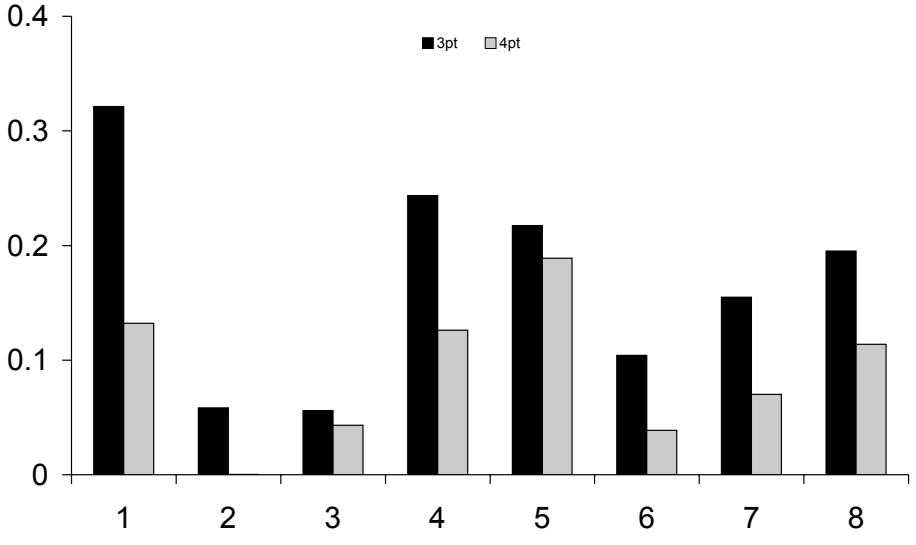


Fig. 4. Normal angle errors (degrees) for 4 test curves (11 points)

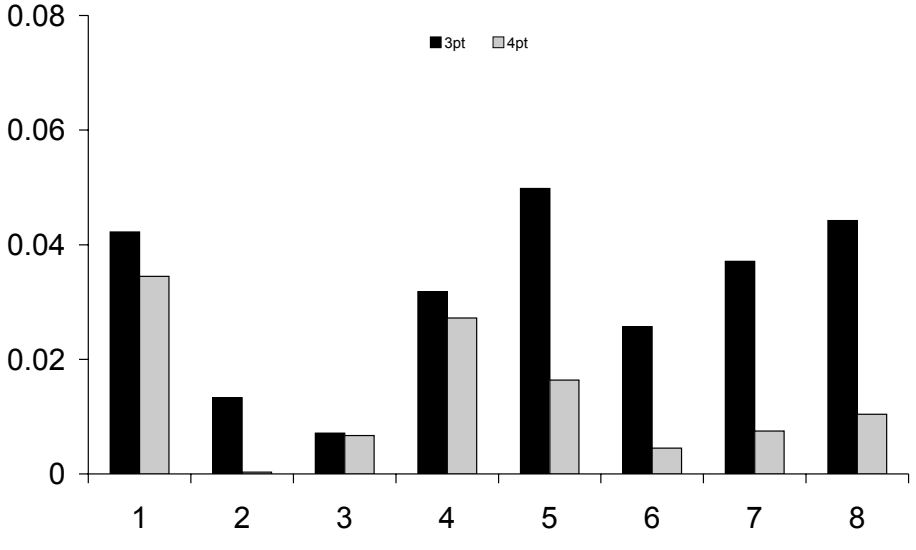


Fig. 5. Normal angle errors (degrees) for 4 test curves (21 points)

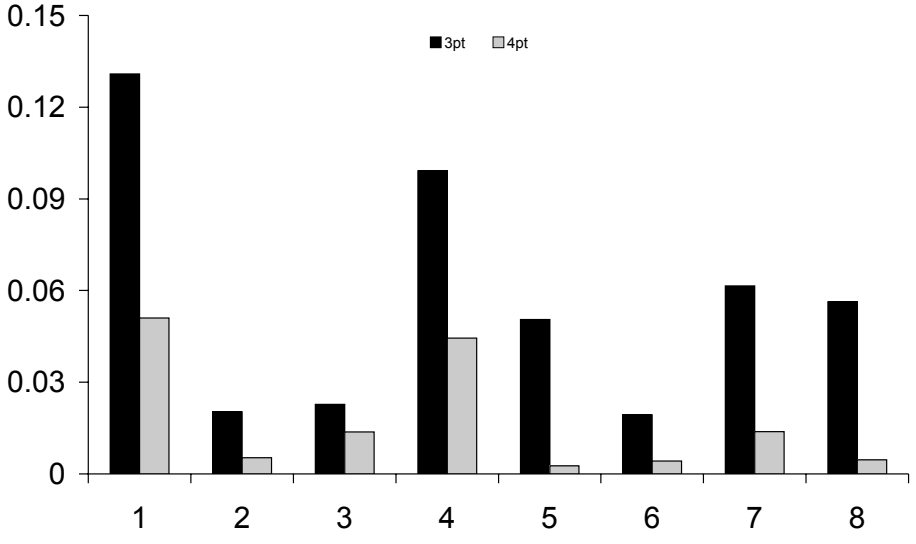


Fig. 6. Curvature errors for 4 test curves (11 points)

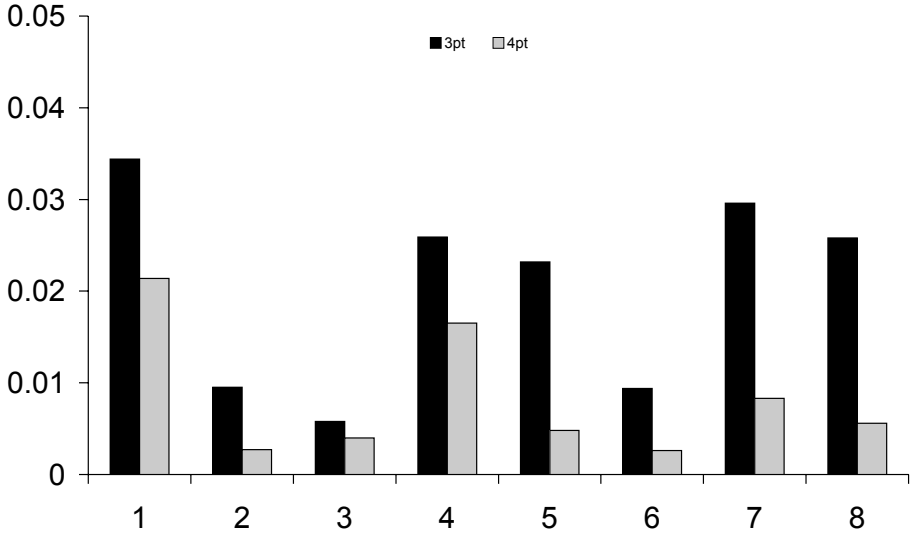


Fig. 7. Curvature errors for 4 test curves (21 points)

comparison. Each curve is interrogated at regularly-spaced parametric intervals to produce irregularly spaced data points with respect to arc-length. At each end point, the normal angle errors and curvature errors between the exact and approximate curvature vectors using the two estimation methods are compared.

The selected test curves are: parabolic arc, exponential function, trigonometric function and elliptic arc. The parametric curves are generally expressed by

$$\mathbf{r}(t) = (x(t), y(t)) \qquad t \in [0, 1],$$

where $t \in [0, 1]$ is chosen in order to generate curves with different end curvature vectors enabling a more critical assessment of the quality of the estimation. Eleven equally spaced points are generated from each curve according to the sampling strategy given in [15]. The first four and last four points are used to estimate the curvature vectors at $\mathbf{r}_0$ and $\mathbf{r}_n$ respectively. To investigate the stability of the estimates, the parameter interval is sub-divided giving 21 points and the estimation is repeated. It is expected that the estimates will improve if the algorithm is well behaved. The arc-lengths used in the estimation are approximated by circular spans.

Fig. 4 illustrates the normal angle errors when 11 points are sampled. The dark columns represent the 3-point method and the grey columns represent the 4-point method. The first 4 columns are the errors at $\mathbf{r}_0$ for the parabolic, exponential, trigonometric and elliptic curves respectively. The last 4 columns are the errors at $\mathbf{r}_n$. Similarly, Fig. 5 illustrates the normal angle errors when 21 points are sampled. The curvature errors at $\mathbf{r}_0$ and $\mathbf{r}_n$ for 11 and 21 sampled points are given in Figs. 6 and 7 respectively.

In all cases, the 4-point method gives smaller errors than the 3-point method (Figs. 4–7). When the parameter interval is subdivided, the errors for both methods become smaller with the 4-point method again performing better than the 3-point. This suggests that the proposed estimates are reliable under the same character of geometry with different spacing strategy, since the estimates tend to their limit values when the parameter interval decreases.

7 Conclusions

Improved 4-point estimates of the curvature vectors at the end points of a string of planar points have been proposed. The curvature vector estimates using the 4-point method are more accurate than those using the 3-point method under all the test configurations. The improvement is expected since the estimates were constructed using purely geometric error analysis and the application of Richardson's extrapolation principle. The accuracy of the estimates is dependent on the arc lengths the derivatives of curvature with respect to s. Although the estimates are most appropriate for points whose high order derivatives of curvature can be ignored, experimental evidence suggests that they can still be used for any curvature behaviour. By expressing the 3^{rd} and 4^{th} order derivatives of a space curve in terms of unit tangent and normal vectors, the estimation formulae presented in this paper could be extended to non-planar points. Finally, consideration needs to be given to the performance of the proposed 4-point estimates when applied to 'rough' or 'noisy' data.

Acknowledgments. The authors would like to thank Prof. Alan A. Ball for valuable suggestions regarding the paper. The authors are pleased to acknowledge the financial support of the Overseas Research Student Awards Scheme of UK (No. 2002005034) and the School of Mechanical Engineering of the University of Birmingham.

References

1. Akima, A.: A new method of interpolation and smooth curve fitting based on local procedures. Journal of the Association for Computing Machinery 17(4), 589–602 (1970)
2. Anoshkina, E., Belyaev, A., Seidel, H.-P.: Asymptotic Analysis of Three-point Approximation of Vertex Normals and Curvatures. In: Proc. Vision, Modelling and Visualisation, Erlangen, Germany, pp. 211–216 (2002)
3. Ball, A.A.: CAD: master or servant of engineering? The Mathematics of Surfaces VII. Information Geometers, Winchester, pp. 17–23 (1997)
4. Faires, J.D.: Numerical Methods. Pacific Grove, London (1998)
5. Farin, G.E.: Curves and Surfaces for Computer Aided Geometric Design—A Practical Guide, 3rd edn. Academic Press, Boston (1993)
6. Faux, I.D., Pratt, M.J.: Computational Geometry for Design and Manufacturing. Ellis Horwood, West Sussex (1979)
7. Gray, A.: Modern Differential Geometry of Curves and Surfaces. CRC Press, Florida (1993)
8. Hamann, B., Chen, J.-L.: Data point selection for piecewise linear curve approximation. Computer Aided Geometric Design 11, 289–301 (1994)
9. Lewiner, T., Gomes, J.D., Lopes, H., Craizer, M.: Curvature and torsion estimators based on parametric curve fitting. Computer & Graphics 29(5), 641–655 (2005)
10. Mjolsness, R.C., Swartz, B.: Some Planar Curvature Approximations. Mathematics of Computation 49(179), 215–230 (1987)
11. Nutbourne, A.W., Martin, R.R.: Differential Geometry Applied to Curve and Surface Design Foundations, vol. 1. Ellis Horwood, West Sussex (1988)
12. O Neill, E.F.: Geometry based constructions for curves and surfaces. Ph.D. Thesis, The University of Birmingham (1993)
13. Piegl, L.: Interactive data interpolation by rational Bézier curves. IEEE Computer Graphics and Applications, pp. 45–58 (1987)
14. Su, B., Liu, D.-Y.: Computational Geometry. Academic Press, San Diego (1989)
15. Tookey, R.M., Ball, A.A.: Estimation of curvatures from planar point data. The Mathematics of Surfaces VII, Information Geometers, Winchester, pp. 131–144 (1997)

Appendix: Derivation of $\mathbf{k}_a$

Let $\mathbf{r}', \mathbf{r}'', \mathbf{r}'''$ and $\mathbf{r}''''$ express derivatives of $\mathbf{r}(s)$ with respect to arc-length s. $\mathbf{r}'$ is a unit tangent vector, so

$$\|\mathbf{r}'\|^2 = \mathbf{r}' \cdot \mathbf{r}' = 1. \tag{1}$$

Differentiating with respect to arc-length three times gives

$$\begin{aligned}\mathbf{r}' \cdot \mathbf{r}'' &= 0 \\ \mathbf{r}' \cdot \mathbf{r}''' &= -\mathbf{r}'' \cdot \mathbf{r}'' \\ \mathbf{r}' \cdot \mathbf{r}'''' &= -3\mathbf{r}'' \cdot \mathbf{r}'''.\end{aligned} \tag{2}$$

The curvature vector $\mathbf{k}_a$ at $\mathbf{r}_0$ of the interpolating circle through points $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2\}$ given in Section 3 by Eq.(12) can be rearranged as:

$$\mathbf{k}_a = \frac{2}{\|\mathbf{C}\|^2}\left(\frac{(\mathbf{A}\cdot\mathbf{C})\,\mathbf{A}}{\|\mathbf{A}\|^2} - \frac{(\mathbf{B}\cdot\mathbf{C})\,\mathbf{B}}{\|\mathbf{B}\|^2}\right). \tag{3}$$

Suppose $\mathbf{r}(s)$ can be expanded by its Taylor series, then $\mathbf{r}(s+\delta s)$ is given by

$$\mathbf{r}(s+\delta s) = \mathbf{r}(s) + \delta s\mathbf{r}' + \frac{\delta s^2}{2}\mathbf{r}'' + \frac{\delta s^3}{6}\mathbf{r}''' + \frac{\delta s^4}{24}\mathbf{r}'''' + O^5, \tag{4}$$

where O^5 denotes all terms of order five or higher. Since $\mathbf{r}_0 = \mathbf{r}(s)$, substituting from the Taylor series of $\mathbf{r}_1 = \mathbf{r}(s+s_0)$ and $\mathbf{r}_2 = \mathbf{r}(s+s_1)$ gives

$$\mathbf{A} = s_0\left(\mathbf{r}' + \frac{s_0}{2}\mathbf{r}'' + \frac{s_0^2}{6}\mathbf{r}'' + \frac{s_0^3}{24}\mathbf{r}''''\right) + O^5, \tag{5}$$

$$\mathbf{B} = s_1\left(\mathbf{r}' + \frac{s_1}{2}\mathbf{r}'' + \frac{s_1^2}{6}\mathbf{r}'' + \frac{s_1^3}{24}\mathbf{r}''''\right) + O^5, \tag{6}$$

$$\begin{aligned}\mathbf{C} =& (s_0 - s_1)(\mathbf{r}' + \frac{(s_0+s_1)}{2}\mathbf{r}'' + \frac{(s_0^2 + s_0s_1 + s_1^2)}{6}\mathbf{r}''' \\ &+ \frac{(s_0+s_1)(s_0^2+s_1^2)}{24}\mathbf{r}'''') + O^5.\end{aligned} \tag{7}$$

4^{th} order derivatives have been included to obtain an estimate of curvature vector for the case of three collinear points. Substituting from Eqs.(5,7) gives

$$\begin{aligned}\mathbf{A}\cdot\mathbf{C} =& s_0(s_0 - s_1)(\mathbf{r}'\cdot\mathbf{r}' + \frac{(2s_0+s_1)}{2}\mathbf{r}'\cdot\mathbf{r}'' \\ &+ \frac{(2s_0^2 + s_0s_1 + s_1^2)}{6}\mathbf{r}'\cdot\mathbf{r}''' + \frac{(2s_0^2 + s_0s_1^3 + s_0^2s_1 + s_1^3)}{24}\mathbf{r}'\cdot\mathbf{r}'''' \\ &+ \frac{(s_0^2 + s_0s_1)}{4}\mathbf{r}''\cdot\mathbf{r}'' + \frac{(2s_0^3 + 2s_0^2s_1 + s_0s_1^2)}{12}\mathbf{r}''\cdot\mathbf{r}''') + O^6.\end{aligned} \tag{8}$$

Substituting from Eqs.(1,2) gives

$$\begin{aligned}\mathbf{A}\cdot\mathbf{C} =& s_0(s_0 - s_1)(1 - \frac{(s_0^2 - s_0s_1 + 2s_1^2)}{12}\mathbf{r}''\cdot\mathbf{r}'' \\ &- \frac{(2s_0^3 - s_0^2s_1 + s_0s_1^2 + 3s_1^3)}{24}\mathbf{r}''\cdot\mathbf{r}''') + O^6.\end{aligned} \tag{9}$$

Similarly

$$\begin{aligned}\mathbf{B}\cdot\mathbf{C} =& s_1(s_0-s_1)(1-\frac{(s_1^2-s_0s_1+2s_0^2)}{12}\mathbf{r}''\cdot\mathbf{r}''\\ &-\frac{(2s_1^3-s_1^2s_0+s_1s_0^2+3s_0^3)}{24}\mathbf{r}''\cdot\mathbf{r}''')+O^6.\end{aligned} \tag{10}$$

$$\mathbf{A}\cdot\mathbf{A}=\|\mathbf{A}\|^2=s_0^2\left(1-\frac{s_0^2}{12}\mathbf{r}''\cdot\mathbf{r}''-\frac{s_0^3}{12}\mathbf{r}''\cdot\mathbf{r}'''\right)+O^6. \tag{11}$$

Now

$$\frac{1}{\|\mathbf{A}\|^2}\approx\frac{1}{s_0^2}\frac{1}{\left(1-\frac{s_0^2}{12}\mathbf{r}''\cdot\mathbf{r}''-\frac{s_0^3}{12}\mathbf{r}''\cdot\mathbf{r}'''\right)}\approx\frac{1}{s_0^2}\left(1+\frac{s_0^2}{12}\mathbf{r}''\cdot\mathbf{r}''+\frac{s_0^3}{12}\mathbf{r}''\cdot\mathbf{r}'''\right) \tag{12}$$

ignoring terms of O^4. Similarly,

$$\frac{1}{\|\mathbf{B}\|^2}\approx\frac{1}{s_1^2}\left(1+\frac{s_1^2}{12}\mathbf{r}''\cdot\mathbf{r}''+\frac{s_1^3}{12}\mathbf{r}''\cdot\mathbf{r}'''\right) \tag{13}$$

$$\frac{1}{\|\mathbf{C}\|^2}\approx\frac{1}{(s_0-s_1)^2}\left(1+\frac{(s_0-s_1)^2}{12}\mathbf{r}''\cdot\mathbf{r}''+\frac{(s_0-s_1)^2(s_0+s_1)}{12}\mathbf{r}''\cdot\mathbf{r}'''\right). \tag{14}$$

Substituting from Eqs.(9,12) gives

$$\frac{\mathbf{A}\cdot\mathbf{C}}{\|\mathbf{A}\|^2}=\frac{(s_0-s_1)}{s_0}\left(1+\frac{(s_0s_1-2s_1^2)}{12}\mathbf{r}''\cdot\mathbf{r}''+\frac{(s_0^2s_1-s_0s_1^2-3s_1^3)}{24}\mathbf{r}''\cdot\mathbf{r}'''\right)+O^4. \tag{15}$$

Substituting from Eqs.(5,15) gives

$$\begin{aligned}\frac{(\mathbf{A}\cdot\mathbf{C})\,\mathbf{A}}{\|\mathbf{A}\|^2}=&(s_0-s_1)(\mathbf{r}'+\frac{s_0}{2}\mathbf{r}''+\frac{s_0^2}{6}\mathbf{r}'''+\frac{s_0^3}{24}\mathbf{r}''''+\frac{(s_0s_1-2s_1^2)}{12}(\mathbf{r}''\cdot\mathbf{r}'')\mathbf{r}'\\ &+\frac{(s_0^2s_1-2s_0s_1^2)}{24}(\mathbf{r}''\cdot\mathbf{r}'')\mathbf{r}''+\frac{(s_0^2s_1-s_0s_1^2-3s_1^3)}{24}\mathbf{r}''\cdot\mathbf{r}''')\mathbf{r}')+O^4.\end{aligned} \tag{16}$$

Similarly,

$$\begin{aligned}\frac{(\mathbf{B}\cdot\mathbf{C})\,\mathbf{B}}{\|\mathbf{B}\|^2}=&(s_0-s_1)(\mathbf{r}'+\frac{s_1}{2}\mathbf{r}''+\frac{s_1^2}{6}\mathbf{r}'''+\frac{s_1^3}{24}\mathbf{r}''''+\frac{(s_0s_1-2s_0^2)}{12}(\mathbf{r}''\cdot\mathbf{r}'')\mathbf{r}'\\ &+\frac{(s_0s_1^2-2s_0^2s_1)}{24}(\mathbf{r}''\cdot\mathbf{r}'')\mathbf{r}''+\frac{(s_1^2s_0-s_1s_0^2-3s_0^3)}{24}(\mathbf{r}''\cdot\mathbf{r}''')\mathbf{r}')+O^4.\end{aligned} \tag{17}$$

Finally, substituting Eqs.(14,16,17) into (3) gives

$$\begin{aligned}\mathbf{k}_a =&\mathbf{r}''(s)+\frac{(s_0+s_1)}{3}\left((\mathbf{r}''(s)\cdot\mathbf{r}''(s))+\mathbf{r}'''(s)\right)\\ &+\frac{(3s_0^2+5s_0s_1+3s_1^2)}{12}\left((\mathbf{r}''(s)\cdot\mathbf{r}'''(s))\mathbf{r}'(s)\right)\\ &+\frac{(s_0^2+s_0s_1+s_1^2)}{12}\left((\mathbf{r}''(s)\cdot\mathbf{r}''(s))\mathbf{r}''(s)+\mathbf{r}''''(s)\right)+O^3.\end{aligned}$$

Inversion, Degree and Reparametrization for Rational Surfaces

Sonia Pérez-Díaz

Dpto de Matemáticas, Universidad de Alcalá. E-28871 Madrid, Spain
sonia.perez@uah.es

Abstract. Let $\mathcal{P}$ be a rational affine parametrization of an algebraic surface $\mathcal{V}$, and let $\phi_{\mathcal{P}} : \mathbb{K}^2 \longrightarrow \mathcal{V}; \overline{t} \longmapsto \mathcal{P}(\overline{t})$ be the rational map induced by $\mathcal{P}$. In this survey, we consider three different problems. First we deal with the problem of deciding whether $\phi_{\mathcal{P}}$ is birational (i.e. whether $\mathcal{P}$ is proper); in case of birationality, the question of computing the inverse of the parametrization is considered. On the other side, the birationality of $\phi_{\mathcal{P}}$ is also characterized by $\deg(\phi_{\mathcal{P}}) = 1$. Hence the problem of analyzing the birationality is equivalent to computing $\deg(\phi_{\mathcal{P}})$. The second problem considered deals with this question. More precisely, we show that $\deg(\phi_{\mathcal{P}})$ can be computed by means of greatest common divisor (gcd) and univariate resultant computations. Finally, if the given parametrization $\mathcal{P}$ is not proper and satisfies an additional condition, we solve the problem of proper reparametrization. That is, we determine a proper rational parametrization $\mathcal{Q}(\overline{t})$ of $\mathcal{V}$ from $\mathcal{P}$ such that $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$. All the results in this survey are included in Perez-Diaz et al. (2002), Perez-Diaz and Sendra (2004) or Perez-Diaz (2006).

1 Introduction

Let $\mathcal{P}(t_1, t_2)$ be a rational affine parametrization of a unirational surface $\mathcal{V}$ over an algebraically closed field $\mathbb{K}$ of characteristic zero. Associated with $\mathcal{P}$, we have the rational map $\phi_{\mathcal{P}} : \mathbb{K}^2 \longrightarrow \mathcal{V}; (t_1, t_2) \longmapsto \mathcal{P}(t_1, t_2)$, where $\phi_{\mathcal{P}}(\mathbb{K}^2) \subset \mathcal{V}$ is dense. $\phi_{\mathcal{P}}$ induces, over the fields of rational functions, the monomorphism $\phi^*_{\mathcal{P}} : \mathbb{K}(\mathcal{V}) \longrightarrow \mathbb{K}(t_1, t_2); R(x_1, x_2, x_3) \longmapsto R(\mathcal{P}(t_1, t_2))$. Then, the degree of the rational map $\phi_{\mathcal{P}}$ is defined as the degree of the finite field extension $\phi^*_{\mathcal{P}}(\mathbb{K}(\mathcal{V})) \subset \mathbb{K}(\overline{t})$; that is $\deg(\phi_{\mathcal{P}}) = [\mathbb{K}(t_1, t_2) : \phi^*_{\mathcal{P}}(\mathbb{K}(\mathcal{V}))]$. In fact, $\deg(\phi_{\mathcal{P}})$ is the cardinality of a generic fibre of $\phi_{\mathcal{P}}$, and in particular the birationality of $\phi_{\mathcal{P}}$ is characterized by $\deg(\phi_{\mathcal{P}}) = 1$ (see Harris (1995) and Shafarevich (1994)).

In this situation, the following three problems are considered:

(1) Decide whether $\mathcal{P}(t_1, t_2)$ is proper, i.e. $\phi_{\mathcal{P}}$ is invertible, and in the affirmative case, compute its rational inverse (inversion problem).
(2) Compute the degree of $\phi_{\mathcal{P}}$, that is, how often the parametrization traces the image of $\phi_{\mathcal{P}}$ (degree problem).
(3) Finally, if the parametrization is not proper, the question of reparametrizing such a surface as to make it properly parametrized is considered (proper reparametrization problem).

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 320–340, 2007.
© Springer-Verlag Berlin Heidelberg 2007

These problems are relevant to some applications in computer aided geometric design (CAGD) where objects are often manipulated parametrically. In particular, proper parametrizations are important in many practical applications, such as visualization (Hoffmann et al. (1997) and Hoschek and Lasser (1993)), rational parametrization of offsets (Arrondo et al. (1997)) and implicitization using resultants (see Cox et al. (1998), Sendra and Winkler (2001)).

The notion of degree appears in some applied and computational problems, for instance, in plotting since the degree measures how often the parametrization traces the image. Also, when computing the implicit equation of a surface by means of resultants, the degree appears as the power of the defining polynomial (see Chionh and Goldman (1992)). In addition, in Perez-Diaz and Sendra (2005), we present formulae for the computation of the partial degrees w.r.t. each variable of the implicit equation of a rational surface defined by a rational parametrization. These formulae are based on the computation of the degree of the rational maps induced by the projections, onto the coordinate planes of the three dimensional space, of the input surface parametrization. Moreover, the birationality of $\phi_{\mathcal{P}}$ is characterized by $\deg(\phi_{\mathcal{P}}) = 1$, and therefore it provides a method for testing the rationality of $\mathcal{V}$ or the properness of $\mathcal{P}$.

In Perez-Diaz et al. (2002), we present a criterion for deciding the correctness of rational parametrizations of hypersurfaces, based on the existence of intersection points of some algebraic hypersurfaces over the field of rational functions $\mathbb{K}(\mathcal{V})$ of the given hypersurface. For the case of surfaces, these auxiliary hypersurfaces are plane curves, and therefore the problem is solved by means of resultants. As a consequence of these results, we present an algorithm that decides whether a given rational parametrization is proper, and in the affirmative case, we compute the inverse. An algorithmic approach solving this problem based on Gröbner Basis is in Schicho (1998). This question is also solved by using multiresultants in Chionh and Goldman (1992).

The more general problem of determining the degree of a rational map can be approached also by means of elimination theoretic methods (see e.g. Chionh and Goldman (1992)). For the special case of rational curves, in the paper by Sendra and Winkler (2001) a method for computing the degree by means of gcds is presented. Moreover, it is also shown how $\deg(\phi_{\mathcal{P}})$ is related to the degree of the rational parametrization; that is, to the maximum degree of the rational components in the parametrization. This fact provides a nice characterization for the birationality in terms of the degrees of the rational components of $\mathcal{P}(t)$ and the partial degrees of the implicit equation of the curve (see Sendra and Winkler (2001)). In Perez-Diaz and Sendra (2004), we deal with the problem for the case of surfaces. More precisely, we prove that the degree of the induced rational map of a rational surface parametrization can be expressed as the degree of a polynomial that is directly computed from the parametrization by means of gcds and univariate resultants. One proves that the cardinality of the fiber, i.e. the degree of $\phi_{\mathcal{P}}$, is the degree of a polynomial which roots correspond to the first coordinate of the fiber element, and that is computed as

the content of a univariate resultant. These ideas are also applied to the explicit computation of the elements of the fibre.

Finally, the question of reparametrizing a surface so as to make it properly parametrized is partially solved in Perez-Diaz (2006). The basic idea is to compute two univariate resultants of certain curves directly constructed from the given parametrization. No results approaching the problem algorithmically are known currently. For the case of curves, it is always possible to reparametrize an improperly parametrized curve in such a way that it becomes properly parametrized. In Perez-Diaz (2006), we also deal with the problem for the case of curves, and we present a new approach to compute a proper parametrization from a given improper one. This new method improves and reinterprets some previous results (see Alonso et al. (1995), Gutierrez et al. (2002) or Sederberg (1986)).

The structure of the paper is as follows: In Section 2 we state the characterization of the properness of rational parametrizations of surfaces, and we provide an algorithmic approach. Section 3 is devoted to the problem of computing the degree of a rational map. For this purpose several results are stated. First, we show how to compute the elements in the fiber, and secondly we show how to compute the cardinality of the fiber when working within a suitable nonempty Zariski open subset of $\mathbb{K}^2$. Finally, we adapt these results to compute the cardinality of the fibre without computing explicitly its elements. We also present the algorithms and examples. Section 4 focuses on the problem of proper reparametrization for improperly parametrized surfaces. In particular, we show how to solve the problem for some parametrizations of special surfaces. We outline the algorithm, and we illustrate it with some examples.

2 Inversion of a Rational Surface Parametrization

In Perez-Diaz et al. (2002) a characterization of the correctness of rational parametrizations of hypersurfaces is presented. The idea consists of analyzing the existence of intersection points of some additional algebraic hypersurfaces directly generated from the given parametrization. For the case of surfaces, it is shown how these results provide an algorithmic criterion that decides whether a given rational parametrization is proper. The algorithm also computes the inverse of the parametrization. For surfaces, the auxiliary hypersurfaces are plane curves, and hence the algorithm is essentially based on resultants.

In this section, we summarize the results in Perez-Diaz et al. (2002) for the case of surfaces.

For this purpose, through this section, we consider $\mathbb{K}$ to be an algebraically closed field of characteristic zero. In addition, let $\mathcal{V}$ be an affine rational surface, and if $\mathcal{P}(t_1, t_2)$ is a rational affine parametrization of $\mathcal{V}$ over $\mathbb{K}$, we write its components as

$$\mathcal{P}(\overline{t}) = \left(\frac{p_{11}(\overline{t})}{p_{12}(\overline{t})}, \frac{p_{21}(\overline{t})}{p_{22}(\overline{t})}, \frac{p_{31}(\overline{t})}{p_{32}(\overline{t})} \right) \in \mathbb{K}(\overline{t})^3,$$

where $\overline{t} = (t_1, t_2)$, and $\gcd(p_{i1}, p_{i2}) = 1$ for $i = 1, 2, 3$. Finally, $F \in \mathbb{K}[x_1, x_2, x_3]$ denotes an irreducible polynomial defining the rational surface $\mathcal{V}$.

We consider K the algebraic closure of the field $\mathbb{K}(\mathcal{V})$; i.e $K = \overline{\mathbb{K}(\mathcal{V})}$, and V_1, V_2, V_3 the plane curves defining, respectively, over K, by the polynomials

$$G_j^{\mathcal{P}}(\overline{t}) = x_j p_{j2}(\overline{t}) - p_{j1}(\overline{t}), \quad j = 1, 2, 3.$$

Moreover, we introduce the curve V_4 that is defined over K by the polynomial

$$G_4^{\mathcal{P}}(\overline{t}) = \operatorname{lcm}(p_{21}(\overline{t}), p_{22}(\overline{t}), p_{23}(\overline{t})).$$

For simplicity, we assume that $\mathcal{P}(\overline{t})$ does not parametrize a plane. Note that the problem for the case of planes is trivial. Note that, one can check whether $\mathcal{P}(\overline{t})$ corresponds to a plane by introducing undermined coefficients a, b, c, d and solving the linear system of equations generated by the equality

$$a\frac{p_{11}(\overline{t})}{p_{12}(\overline{t})} + b\frac{p_{21}(\overline{t})}{p_{22}(\overline{t})} + c\frac{p_{31}(\overline{t})}{p_{32}(\overline{t})} = d.$$

In these conditions, we get the following theorem that characterizes the properness of $\mathcal{P}$ by means of the existence of the intersection points of the curves V_1, V_2, V_3, V_4.

Theorem 1. *The following statements are equivalent:*

1. *The parametrization $\mathcal{P}(\overline{t})$ is proper.*
2. *There exists exactly one point $A = (A_1, A_2) \in (V_1 \cap V_2 \cap V_3) \cap (K^2 \setminus \mathbb{K}^2)$.*
3. *There exists exactly one point $A = (A_1, A_2) \in (V_1 \cap V_2 \cap V_3) \cap (K \setminus \mathbb{K})^2$.*
4. *There exists exactly one point $A = (A_1, A_2) \in (V_1 \cap V_2 \cap V_3) \setminus (V_1 \cap V_2 \cap V_3 \cap V_4)$.*

Furthermore, if either Statement 2 or 3 or 4 holds then A is the inverse of $\mathcal{P}$.

In the following, we show how Statements 2, 3 and 4 of Theorem 1 can be decided computationally. For this purpose, in the following technical result, we state a one to one relation between the intersection points of any three plane curves, without common components, and the roots of a univariate polynomial computed by a resultant.

Proposition 1. *Let $\mathcal{L}$ be a subfield of an algebraically closed field $\mathbb{F}$ of characteristic zero, and let C_1, C_2, C_3 be plane algebraic curves over $\mathbb{F}$, with no common components, defined by the polynomials F_1, F_2, $F_3 \in \mathcal{L}[t_1, t_2]$, respectively. Let F_1, F_2, F_3 be such that each two of their leading coefficients, w.r.t. one of the variables, have a trivial gcd. Let F_1 not have a factor in $\mathcal{L}[t_1]$. Then, the t_1-coordinates of the intersection points of C_1, C_2, C_3 are the roots of the content w.r.t Z of the resultant w.r.t. t_2 of the polynomials $F_1, F_2 + ZF_3$.*

One may check that our auxiliary curves V_1, V_2, V_3 are in the conditions of Proposition 1. That is, the curves V_1, V_2, V_3, do not have common components, and at least one of the polynomials $G_1^{\mathcal{P}}, G_2^{\mathcal{P}}, G_3^{\mathcal{P}}$ does not have factors in $\mathbb{K}(\mathcal{V})[t_1]$

neither in $\mathbb{K}(\mathcal{V})[t_2]$ (see Propositions 2 and 3 in Perez-Diaz et al. (2002)). In addition, we assume w.l.o.g that each two of the leading coefficients of $G_1^{\mathcal{P}}, G_2^{\mathcal{P}}, G_3^{\mathcal{P}}$, w.r.t. t_2 is trivial, (resp. w.r.t. t_1). If not, we consider a suitable change of coordinates.

Thus, one can derive an algorithmic method to compute the t_1 and t_2 coordinates of the points in $V_1 \cap V_2 \cap V_3$. We state the result for the t_1–coordinates, and we assume w.l.o.g. that $G_1^{\mathcal{P}}$ is the polynomial without factors in $\mathbb{K}(\mathcal{V})[t_1]$.

Theorem 2. *The t_1-coordinates of the intersection points of V_1, V_2, V_3 are the roots of the polynomial*

$$S(t_1) = \text{Content}_Z(\text{Res}_{t_2}(G_1^{\mathcal{P}}, G_2^{\mathcal{P}} + ZG_3^{\mathcal{P}})) \in \mathbb{K}(\mathcal{V})[t_1],$$

where Content_Z *denotes the content w.r.t. a new variable Z, and* $\text{Res}_{t_2}(G_1^{\mathcal{P}}, G_2^{\mathcal{P}} + ZG_3^{\mathcal{P}})$ *is the resultant of the polynomials $G_1^{\mathcal{P}}, G_2^{\mathcal{P}} + ZG_3^{\mathcal{P}}$ w.r.t. t_2.*

A similar result holds for the t_2–coordinates. In this case, we denote by $T(t_2)$ the polynomial defining the t_2-coordinates of the intersection points of V_1, V_2, V_3. That is,

$$T(t_2) = \text{Content}_Z(\text{Res}_{t_1}(G_1^{\mathcal{P}}, G_2^{\mathcal{P}} + ZG_3^{\mathcal{P}})) \in \mathbb{K}(\mathcal{V})[t_2].$$

Taking into account Theorem 1, we observe that we are interested in computing the defining polynomial of the t_1-coordinates in $(V_1 \cap V_2 \cap V_3) \cap (K \setminus \mathbb{K})^2$. For this purpose, in the following we consider the polynomials $\overline{S}(t_1)$ obtained by crossing out the constant roots of the polynomial $S(t_1)$, and $\overline{T}(t_2)$ obtained by crossing out the constant roots of the polynomial $T(t_2)$.

In these conditions, one deduces the following characterization.

Theorem 3. *Let $\overline{S}(t_1)$ and $\overline{T}(t_2)$ be the defining polynomials of the t_1-coordinates and the t_2-coordinates of the points in $(V_1 \cap V_2 \cap V_3) \cap (K \setminus \mathbb{K})^2$, respectively. The following statements are equivalent:*

1. *$\mathcal{P}$ is proper.*
2. *$\overline{S}(t_1)$ is linear. Furthermore, if $A_1 \in K$ is the root of $\overline{S}(t_1)$, then the polynomial*

$$M(t_2) = \frac{\gcd_{\mathbb{K}(V)[t_2]}(G_1^{\mathcal{P}}(A_1, t_2), G_2^{\mathcal{P}}(A_1, t_2), G_3^{\mathcal{P}}(A_1, t_2))}{\gcd_{\mathbb{K}(\mathcal{V})[t_2]}(G_1^{\mathcal{P}}(A_1, t_2), G_2^{\mathcal{P}}(A_1, t_2), G_3^{\mathcal{P}}(A_1, t_2), G_4^{\mathcal{P}}(A_1, t_2))}$$

   only has one distinct root.
3. *$\overline{S}(t_1)$ and $\overline{T}(t_2)$ are linear.*

2.1 Algorithm and Examples

The results presented above can be applied to derive an algorithm that decides the correctness of a given rational parametrization of a surface, and that (in the affirmative case) determines the inverse. In this subsection, we outline the algorithm and we illustrate it by examples. The algorithm is given in Figure 1.

Algorithm Inversion.

Given $F \in \mathbb{K}[x,y,z]$ irreducible and defining a surface (not being a plane) $\mathcal{V}$ over $\mathbb{K}$, and $\mathcal{P}(\overline{t}) = (\frac{p_{11}}{p_{12}}, \frac{p_{21}}{p_{22}}, \frac{p_{31}}{p_{32}})$ a rational parametrization of $\mathcal{V}$, where $\gcd(p_{i1}, p_{i2}) = 1 \ \forall\, i \in \{1,2,3\}$, the algorithm decides whether $\mathcal{P}$ is proper, and in the affirmative case it computes the inverse of $\mathcal{P}$.

1. Compute the polynomials
$$G_j^{\mathcal{P}}(\overline{t}) = x_j p_{j2}(\overline{t}) - p_{j1}(\overline{t}), \quad j = 1,2,3.$$
that define the curves V_1, V_2, V_3, respectively.
2. Check whether each two of the leading coefficients of $G_1^{\mathcal{P}}, G_2^{\mathcal{P}}, G_3^{\mathcal{P}}$, w.r.t. t_2 is trivial, (resp. w.r.t. t_1). If not, do a suitable change of coordinates.
3. Compute the polynomial $S(t_1)$ defining the t_1-coordinates of the intersection points of V_1, V_2, V_3 (see Theorem 2).
4. Compute the polynomial $\overline{S}(t_1)$ obtained by crossing out the constant roots of the polynomial $S(t_1)$.
5. If $\overline{S}(t_1)$ is not linear then return "$\mathcal{P}$ `is not proper`". Else let α be the root of $\overline{S}(t_1)$
6. Compute the polynomial $T(t_2)$ defining the t_2-coordinates of the intersection points of V_1, V_2, V_3 (see Theorem 2).
7. Compute the polynomial $\overline{T}(t_2)$ obtained by crossing out the constant roots of the polynomial $T(t_2)$.
8. If $\overline{T}(t_2)$ is not linear then return "$\mathcal{P}$ `is not proper`". Else let β be the root of $\overline{T}(t_2)$.
9. Return "(α, β) `is the inverse of` $\mathcal{P}$".

Fig. 1. Algorithm Inversion

Remark 1. If one uses the implicit equation of the surface to carry out the arithmetic over $\mathbb{K}(\mathcal{V})$, since $I(\mathcal{V}) = (F)$, basic arithmetic on $\mathbb{K}[\mathcal{V}]$ can be carried out by computing polynomial remainders. Therefore the quotient field $\mathbb{K}(\mathcal{V})$ is computable. However, one may avoid the requirement on the implicit equation. For this purpose, elements are represented (not uniquely) as function of polynomials in the variables x_1, x_2, x_3. In order to check zero equality, one may use the parametrization. However, this can be too time consuming. Instead, we test zero-equality by substituting a random point on the surface. The result of this zero test is correct with probability almost one. In addition, if the output is a rational inverse, we also test its correctness by checking it on a randomly chosen point on the surface.

Finally we illustrate the algorithm by two examples. In the first one the parametrization turn to be proper and their inverse is determined. However in the second one the algorithm shows that the given parametrization is not proper.

Example 1. Let $\mathcal{V}$ be the surface defined by the irreducible polynomial

$F(x_1,x_2,x_3) = 2x_2 + 56x_1^2x_2x_3^2 + 32x_1^2x_3^2x_2^2 + 16x_1^6 + 48x_1^2x_3^4 - 32x_1^2x_2^3 + 2x_1^2 - 8x_1^4x_2 + 16x_1^4x_2^2 + 48x_1^4x_3^2 + x_3^2 - 90x_1^2x_2 - 52x_1^2x_3^2 - 24x_3^2x_2 - 71x_1^4 - 15x_2^2 - 96x_1^2x_2^2 + 72x_2^2x_3^2 - -8x_3^4 + 24x_2^3 + 16x_2^2x_3^4 + 32x_2^3x_3^2 + 64x_3^4x_2 + 16x_3^6 + 16x_2^4 \in \mathbb{C}[x_1,x_2,x_3]$.

We consider the rational parametrization of $\mathcal{V}$, defined by $\mathcal{P}(t_1,t_2) =$

$$\left(\frac{t_1(4t_2+4t_1^2t_2^2+1+t_2^2)}{4t_1^2t_2^2+1+t_2^2}, \frac{-2t_2+4t_1^4t_2^2+t_1^2-3t_1^2t_2^2-1-t_2^2}{4t_1^2t_2^2+1+t_2^2}, \frac{4t_1^2t_2^2-1+t_2^2}{4t_1^2t_2^2+1+t_2^2}\right).$$

Let us check whether $\mathcal{P}(\overline{t})$ is a proper parametrization. For this purpose, we apply the algorithm. First we consider the polynomials

$$\begin{aligned} G_1^{\mathcal{P}}(\overline{t}) &= 4x_1t_1^2t_2^2 + x_1 + x_1t_2^2 - 4t_1t_2 - 4t_1^3t_2^2 - t_1 - t_1t_2^2,\\ G_2^{\mathcal{P}}(\overline{t}) &= 4x_2t_1^2t_2^2 + x_2 + x_2t_2^2 + 2t_2 - 4t_1^4t_2^2 - t_1^2 + 3t_1^2t_2^2 + 1 + t_2^2,\\ G_3^{\mathcal{P}}(\overline{t}) &= 4x_3t_1^2t_2^2 + x_3 + x_3t_2^2 - 4t_1^2t_2^2 + 1 - t_2^2. \end{aligned}$$

Doing the computation one has that

$$R(t_1,Z) = \mathrm{Res}_{\mathbb{C}(V)[t_1,Z][t_2]}(G_3^{\mathcal{P}}, G_2^{\mathcal{P}} + ZG_1^{\mathcal{P}}, t_2) =$$

$= (2x_2 + 16x_2t_1^4 + 16x_2^2t_1^4 - 32x_2t_1^6 + x_2^2 + 14x_2t_1^2 + t_1^4 + x_3^2 + 4x_3^2t_1^2 + 2t_1^2 + 16t_1^8 + 8x_2^2t_1^2 - 24t_1^6) + (-16t_1^3x_3^2 - 2x_2t_1 - 16x_2t_1^3 + 2t_1^3 - 16t_1^5 - 32t_1^6x_1 + 2x_2x_1 + 32t_1^7 - 32x_2t_1^5 + 14x_1t_1^2 - 4t_1x_3^2 + 2t_1 + 32x_2t_1^4x_1 + 16x_2x_1t_1^2 + 16t_1^4x_1 + 2x_1)Z + (8x_1^2t_1^2 - 2x_1t_1 + 4x_3^2t_1^2 + 16x_3^2t_1^4 - 32t_1^5x_1 + x_1^2 - 16t_1^3x_1 - 8t_1^4 + 16x_1^2t_1^4 - 3t_1^2 + 16t_1^6)Z^2,$

In addition,
$S(t_1) = \mathrm{Content}_Z(R(t_1,Z)) = (4t_1^2+1)((-8x_1^4 - 16x_1^2x_3^2 + 12x_2x_1^2 + 27x_1^2 - 8x_3^4 + 6x_3^2 - 12x_3^2x_2 + 5x_2 - 1 - 4x_2^2)t_1 + 4x_2x_1^3 + 5x_1^3 - 4x_1x_3^2 + 4x_1x_3^2x_2 - 4x_1x_2^2 - 13x_2x_1 - x_1)$.

Note that S has the constant roots corresponding to the factor $4t_1^2+1$ in $\mathbb{C}$, hence
$\overline{S}(t_1) = (-8x_1^4 - 16x_1^2x_3^2 + 12x_2x_1^2 + 27x_1^2 - 8x_3^4 + 6x_3^2 - 12x_3^2x_2 + 5x_2 - 1 - 4x_2^2)t_1 + 4x_2x_1^3 + 5x_1^3 - 4x_1x_3^2 + 4x_1x_3^2x_2 - 4x_1x_2^2 - 13x_2x_1 - x_1$.

Moreover, since $\overline{S}(t_1)$ is linear in the variable t_1, we obtain the first coordinate of the inverse $\alpha(x_1,x_2,x_3) =$

$$\frac{(4x_2x_1^2 + 5x_1^2 - 4x_3^2 + 4x_3^2x_2 - 4x_2^2 - 13x_2 - 1)x_1}{8x_1^4 + 16x_1^2x_3^2 - 12x_2x_1^2 - 27x_1^2 + 8x_3^4 - 6x_3^2 + 12x_3^2x_2 - 5x_2 + 1 + 4x_2^2}.$$

Reasoning similarly with the variable t_2, we obtain

$\overline{T}(t_2) = -16 - 848x_2 + 128x_2x_3x_1^2 + 256x_2^2x_3x_1^2 - 16x_3 - 384x_3^2 - 896x_2^2 - 624x_1^2 + 128x_2x_1^2 + 320x_3^2x_2 + 512x_1^2x_3^2 + 256x_1^4 + 256x_3^4 - 384x_3^3 - 624x_3x_1^2 - 256x_2^3 + 256x_3^5 - 256x_2^3x_3 - 896x_2^2x_3 + 256x_2^2x_1^2 + 256x_2^2x_3^3 + 256x_2^2x_3^2 - 848x_2x_3 + 320x_2x_3^3 + 256x_1^4x_3 +$

$512x_1^2x_3^3+(2816x_2^2x_3^2+1024x_2^3x_3^2-16-2816x_2^2-688x_3^2+1024x_2x_3^4+448x_3^2x_2-1024x_2x_1^2+1280x_1^2x_3^2-1024x_2^3+1024x_2x_1^2x_3^2-1280x_1^2+704x_3^4-1472x_2)t_2$.

Since $\overline{T}(t_2)$ is linear on the variable t_2, we obtain the second coordinate of the inverse

$$\beta(x_1,x_2,x_3)=\frac{\beta_1(x_1,x_2,x_3)}{\beta_2(x_1,x_2,x_3)},$$

where

$\beta_1(x_1,x_2,x_3)=-(16x_3^4+16x_2^2x_3^2+20x_3^2x_2-24x_3^2+32x_1^2x_3^2-1-53x_2+8x_1^2x_2+16x_1^2x_2^2-39x_1^2-56x_2^2-16x_2^3+16x_1^4)$,

and

$\beta_2(x_1,x_2,x_3)=64x_3^3x_2+44x_3^3-44x_3^2-64x_3^2x_2+176x_2^2x_3+64x_2^3x_3+80x_1^2x_3+64x_2x_1^2x_3+92x_3x_2+x_3-1-92x_2-176x_2^2-80x_1^2-64x_1^2x_2-64x_2^3$.

Thus, applying the algorithm, we conclude that $\mathcal{P}$ is proper and its inverse mapping is:

$$\mathcal{P}^{-1}(t_1,\,t_2)=\,(\alpha,\beta)=\,(\alpha(x_1,x_2,x_3)\,,\,\beta(x_1,x_2,x_3)).$$

Example 2. Let $\mathcal{V}$ be the surface defined by the irreducible polynomial

$$F(x_1,x_2,x_3)=-4x_1x_2-2+4x_2+2x_3+2x_3x_2+2x_1\in\mathbb{C}[x_1,x_2,x_3].$$

We consider the rational parametrization of $\mathcal{V}$:

$$\mathcal{P}(t_1,t_2)=\left(\frac{2t_1^2t_2+t_1^2+4t_2+3}{2t_2+1},\frac{1}{(t_1^2+1)(2t_2+1)},\frac{-(-1+2t_1^2t_2+t_1^2+2t_2)}{2t_2+1}\right).$$

Let us check whether $\mathcal{P}(\,\overline{t}\,)$ is a proper parametrization. For this purpose, we apply the algorithm. First we consider the polynomials

$$\begin{aligned}G_1^{\mathcal{P}}(\,\overline{t}\,)&=2x_1t_2+x_1-2t_1^2t_2-t_1^2-4t_2-3,\\ G_2^{\mathcal{P}}(\,\overline{t}\,)&=-1+2x_2t_1^2t_2+x_2t_1^2+2x_2t_2+x_2,\\ G_3^{\mathcal{P}}(\,\overline{t}\,)&=-1+2t_1^2t_2+t_1^2+2t_2+2x_3t_2+x_3.\end{aligned}$$

Doing the computation one has that

$$R(t_1,Z)=\mathrm{Res}_{\mathbb{C}(V)[t_1,Z][t_2]}(G_3^{\mathcal{P}},G_2^{\mathcal{P}}+ZG_1^{\mathcal{P}},t_2)=$$

$$=4+2x_2+10Z+2t_1^2+6Zt_1^2+2Zx_3-4Zx_1+2x_2t_1^2-2x_1,$$

and

$$S(t_1)=\mathrm{Content}_Z(R(t_1,Z))=10+2x_3-4x_1+6t_1^2.$$

Note that S does not have constant roots, and then $\overline{S}(t_1)=S(t_1)$. Since $\overline{S}(t_1)$ is not linear we conclude that $\mathcal{P}$ is not proper.

3 Computation of the Degree of Rational Surface Parametrizations

A rational parametrization $\mathcal{P}(\overline{t})$ of an algebraic surface $\mathcal{V}$ establishes a rational correspondence of the affine plane with the surface:

$$\phi_{\mathcal{P}} : \mathbb{K}^2 \longrightarrow \mathcal{V} \subset \mathbb{K}^3; \overline{t} \longmapsto \mathcal{P}(\overline{t}).$$

This correspondence is a birational equivalence if the parametrization is proper. So, intuitively speaking, a rational proper parametrization traces the surface once or equivalently $\deg(\phi_{\mathcal{P}}) = 1$, where $\deg(\phi_{\mathcal{P}})$ denotes the degree of the rational map $\phi_{\mathcal{P}}$. More precisely, $\deg(\phi_{\mathcal{P}})$ measures how often the parametrization traces the image and then, the birationality of $\phi_{\mathcal{P}}$ is characterized by $\deg(\phi_{\mathcal{P}}) = 1$ (see Harris (1995) and Shafarevich (1994)). Moreover, when the degree is 1, the expression of the fibre of a generic element provides the inverse mapping of $\phi_{\mathcal{P}}$ (see Section 2).

In Perez-Diaz and Sendra (2004), the problem of computing the degree of such a rational map is considered. In general, determining the degree of a rational map can be achieved by means of elimination theoretic methods. For the case of curves, it is proved that the degree can be computed by gcd computations (see Sendra and Winkler (2001)), and for surfaces it is shown that the degree of a rational map induced by a surface parametrization can be computed by means of gcd and univariate resultant computations (see Perez-Diaz and Sendra (2004)).

In this section we survey these results. The basic idea is to express the elements of a generic fibre as the finitely many intersection points of certain curves directly constructed from the parametrization, and defined over the algebraic closure of a field of rational functions.

3.1 Preliminaries and Terminology

In this subsection, we introduce the notation and assumptions that will be used throughout Sections 3 and 4.

First, we consider $\mathbb{K}$ to be an algebraically closed field of characteristic zero, and $\mathbb{K}^\star = \mathbb{K} \setminus \{0\}$. In addition, let $\mathcal{V}$ be an affine rational surface, and if $\mathcal{P}(t_1, t_2)$ is a rational affine parametrization of $\mathcal{V}$ over $\mathbb{K}$, we write its components as

$$\mathcal{P}(\overline{t}) = \left(\frac{p_{11}(\overline{t})}{p_{12}(\overline{t})}, \frac{p_{21}(\overline{t})}{p_{22}(\overline{t})}, \frac{p_{31}(\overline{t})}{p_{32}(\overline{t})} \right) \in \mathbb{K}(\overline{t})^3,$$

where $\overline{t} = (t_1, t_2)$, and $\gcd(p_{i1}, p_{i2}) = 1$ for $i = 1, 2, 3$.

Since $\mathcal{P}(\overline{t})$ is a surface parametrization, at least two of the component gradients are not parallel. Let us assume w.l.o.g that $\{\nabla\left(\frac{p_{11}}{p_{12}}\right), \nabla\left(\frac{p_{21}}{p_{22}}\right)\}$ are linearly independent as vectors in $\mathbb{K}(\overline{t})^3$. This in particular implies that $\frac{p_{11}}{p_{12}}$, $\frac{p_{21}}{p_{22}}$ are both not constant.

Associated to the parametrization $\mathcal{P}(\overline{t})$ we consider the induced rational map:

$$\phi_{\mathcal{P}} : \mathbb{K}^2 \longrightarrow \mathcal{V} \subset \mathbb{K}^3; \overline{t} \longmapsto \mathcal{P}(\overline{t}),$$

where $\phi_{\mathcal{P}}(\mathbb{K}^2) \subset \mathcal{V}$ is dense. $\phi_{\mathcal{P}}$ induces, over the fields of rational functions, the monomorphism $\phi^*_{\mathcal{P}} : \mathbb{K}(\mathcal{V}) \longrightarrow \mathbb{K}(\overline{t}); R(x_1, x_2, x_3) \longmapsto R(\mathcal{P}(\overline{t}))$. Then, the degree of the rational map $\phi_{\mathcal{P}}$ is defined as the degree of the finite field extension $\phi^*_{\mathcal{P}}(\mathbb{K}(\mathcal{V})) \subset \mathbb{K}(\overline{t})$; that is $\deg(\phi_{\mathcal{P}}) = [\mathbb{K}(\overline{t}) : \phi^*_{\mathcal{P}}(\mathbb{K}(\mathcal{V}))]$ (for further details on this notion see e.g. Shafarevich (1994) pp. 143, or Harris (1995) pp. 80). In fact, $\deg(\phi_{\mathcal{P}})$ is the cardinality of a generic fibre of $\phi_{\mathcal{P}}$ (see Theorem 7, pp. 76 in Shafarevich (1994)). We will use this characterization of the degree for our reasoning. For this purpose, for every $\overline{\alpha} \in \mathbb{K}^2$ such that $\mathcal{P}(\overline{\alpha})$ is defined, we denote by $\mathcal{F}_{\mathcal{P}}(\overline{\alpha})$ the fibre of $\overline{\alpha}$ via $\phi_{\mathcal{P}}$; i.e

$$\mathcal{F}_{\mathcal{P}}(\overline{\alpha}) = \{\overline{t} \in \mathbb{K}^2 \mid \mathcal{P}(\overline{t}) = \mathcal{P}(\overline{\alpha})\}.$$

In addition, let $\overline{s} = (s_1, s_2)$ be a generic element of $\mathbb{K}^2$, and let $\mathbb{F}$ be the algebraic closure of the field $\mathbb{K}(\overline{s})$. We consider $V_1^{\overline{s}}, V_2^{\overline{s}}, V_3^{\overline{s}}$ algebraic sets defining, respectively, over $\mathbb{F}$, by the polynomials

$$H_j^{\mathcal{P}}(\overline{t}, \overline{s}) = p_{j1}(\overline{t})p_{j2}(\overline{s}) - p_{j2}(\overline{t})p_{j1}(\overline{s}) \in \mathbb{K}[\overline{s}][\overline{t}], \text{ for } j = 1, 2, 3.$$

We observe that since we have assumed that the components of $\mathcal{P}(\overline{t})$ are not constant, then $H_j^{\mathcal{P}}$ is not identically zero. Moreover, let V_4 be a new curve defined over $\mathbb{F}$ by the polynomial $H_4^{\mathcal{P}}(\overline{t}) = \mathrm{lcm}(p_{12}, p_{22}, p_{32})$.

Also, we assume that none of the projective curves defined by each of the numerators and denominators of the parametrization components pass through the point at infinity $(0 : 1 : 0)$, where the homogeneous variables are (t_1, t_2, w). Note that the degree of a rational map is multiplicative under composition, and thus the composition of $\phi_{\mathcal{P}}$ and linear transformation will preserve the degree of $\phi_{\mathcal{P}}$. Thus, we can assume w.l.o.g. the previous condition. Also, note that under these conditions, if $\overline{H}_i^{\mathcal{P}}(t_1, t_2, w, s_1, s_2)$ denotes the homogenization of $H_i^{\mathcal{P}}(t_1, t_2, s_1, s_2)$ as a polynomial in $\mathbb{K}[\overline{s}][\overline{t}]$, it holds that $\overline{H}_i^{\mathcal{P}}(0, 1, 0, \overline{s}) \neq 0$ for $i = 1, 2$. This in particular implies that the leading coefficients of $H_i^{\mathcal{P}}$ w.r.t. t_2 only depend on $\overline{s}$. Note, that in these conditions we have that each two of the leading coefficients of $H_1^{\mathcal{P}}, H_2^{\mathcal{P}}, H_3^{\mathcal{P}}$, w.r.t. t_2 is trivial.

Finally, we introduce the polynomials

$$S_1^{\mathcal{P}}(t_1, \overline{s}) = \mathrm{pp}_{\overline{s}}(\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}} + ZH_3^{\mathcal{P}}))),$$

$$S_2^{\mathcal{P}}(t_2, \overline{s}) = \mathrm{pp}_{\overline{s}}(\mathrm{Content}_Z(\mathrm{Res}_{t_1}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}} + ZH_3^{\mathcal{P}}))),$$

where $\mathrm{pp}_{\overline{s}}$ denotes the primitive part w.r.t. $\overline{s}$, and $\mathrm{Content}_Z$ denotes the content w.r.t a new variable Z.

3.2 Computation of the Degree of a Rational Map

We first observe that the fibre $\mathcal{P}^{-1}(\mathcal{P}(\overline{s}))$, i.e. $\mathcal{F}_{\mathcal{P}}(\overline{s}) = \{\overline{t} \in \mathbb{F}^2 \,|\, \mathcal{P}(\overline{t}) = \mathcal{P}(\overline{s})\}$ can be expressed as

$$\mathcal{F}_{\mathcal{P}}(\overline{s}) = \left\{\overline{t} \in \mathbb{F}^2 \;\middle|\; H_i{}^{\mathcal{P}}(\overline{t}, \overline{s}) = 0,\ i \in \{1, 2, 3\},\quad H_4^{\mathcal{P}}(\overline{t}) \neq 0\right\}$$

That is,

$$\mathcal{F}_{\mathcal{P}}(\overline{s}) = (V_1^{\overline{s}} \cap V_2^{\overline{s}} \cap V_3^{\overline{s}}) \setminus (V_1^{\overline{s}} \cap V_2^{\overline{s}} \cap V_3^{\overline{s}} \cap V_4).$$

The algebraic sets $V_i^{\overline{s}}$ satisfy some important properties. First, taking into account Lemmas 10 and 11 in Perez-Diaz and Sendra (2004), we have that

$$(V_1^{\overline{s}} \cap V_2^{\overline{s}} \cap V_3^{\overline{s}}) \setminus (V_1^{\overline{s}} \cap V_2^{\overline{s}} \cap V_3^{\overline{s}} \cap V_4) \subset (V_1^{\overline{s}} \cap V_2^{\overline{s}} \cap V_3^{\overline{s}}) \cap (\mathbb{F} \setminus \mathbb{K})^2.$$

In addition, by Lemmas 1 and 2 in Perez-Diaz and Sendra (2004), we deduce that $V_1^{\overline{s}}$, $V_2^{\overline{s}}$, $V_3^{\overline{s}}$ do not have common components, and that the polynomials $H_1^{\mathcal{P}}(\overline{t}, \overline{s})$, and $H_2^{\mathcal{P}}(\overline{t}, \overline{s}) + ZH_3^{\mathcal{P}}(\overline{t}, \overline{s})$ do not have factors in $\mathbb{F}[t_1]$. Thus, by applying Proposition 1, one gets the next result.

Theorem 4. *The following statements hold:*

1. *For $i = 1, 2$, the polynomials $S_i^{\mathcal{P}}$ define the t_i–coordinates of the points in $\mathcal{F}_{\mathcal{P}}(\overline{s})$.*
2. *$\mathcal{F}_{\mathcal{P}}(\overline{s}) = \{(A, B) \in \mathbb{F}^2 \mid S^{\mathcal{P}}(A, \overline{s}) = 0,\ M_A^{\mathcal{P}}(B, \overline{s}) = 0\}$, where*

$$M_A^{\mathcal{P}}(t_2, \overline{s}) = \mathrm{pp}_{\overline{s}}(\gcd_{\mathbb{F}[t_2]}(H_1^{\mathcal{P}}(A, t_2, \overline{s}), H_2^{\mathcal{P}}(A, t_2, \overline{s}), H_3^{\mathcal{P}}(A, t_2, \overline{s}))).$$

3. *$\mathrm{Card}(\mathcal{F}_{\mathcal{P}}(\overline{s})) = \deg(\phi_{\mathcal{P}}) = \deg_{t_i}({S_i}^{\mathcal{P}}(t_i, \overline{s}))$. In particular, $\mathcal{P}$ is proper if and only if, up to constants in $\mathbb{K}^\star$, it holds that $S_i^{\mathcal{P}} = t_i - s_i$.*

In the following, we show how to compute the cardinality of the fibre when working within a suitable non-empty Zariski open subset of $\mathbb{K}^2$. More precisely, we see how the results in Theorem 4 can be particularized to compute the degree for a particular element $\overline{\alpha} \in \mathbb{K}^2$, where $\mathcal{P}(\overline{\alpha})$ is defined. For this purpose, one considers the polynomials

$$H_j^{\mathcal{P}, \overline{\alpha}}(\overline{t}) = p_{j1}(\overline{t})p_{j2}(\overline{\alpha}) - p_{j2}(\overline{t})p_{j1}(\overline{\alpha}) \in \mathbb{K}[\overline{t}], \text{ for } j = 1, 2, 3,$$

and we note that by Proposition 1, the t_1–coordinates of the intersection points of the polynomials $H_i^{\mathcal{P}, \overline{\alpha}}$, $i = 1, 2, 3$ are given by the polynomial

$$\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P}, \overline{\alpha}},\ H_2^{\mathcal{P}, \overline{\alpha}} + ZH_3^{\mathcal{P}, \overline{\alpha}})).$$

(similarly for the t_2–coordinates).

On the other side Proposition 1 can be also easily extended to more than three curves. Hence, one has that the t_1–coordinates of the intersection points of the polynomials $H_i^{\mathcal{P}, \overline{\alpha}}$, $i = 1, 2, 3, 4$ are given by the polynomial

$$\mathrm{Content}_{\{Z,W\}}(\mathrm{Res}_{t_2}(H_1^{\mathcal{P}, \overline{\alpha}},\ H_2^{\mathcal{P}, \overline{\alpha}} + ZH_3^{\mathcal{P}, \overline{\alpha}} + WH_4^{\mathcal{P}}))$$

(similarly for the t_2–coordinates).

Furthermore, we observe that for every $\overline{\alpha}$ in a non empty subset of $\mathbb{K}^2$, the fibre $\mathcal{P}^{-1}(\mathcal{P}(\overline{\alpha}))$, i.e. $\mathcal{F}_{\mathcal{P}}(\overline{\alpha}) = \{\overline{t} \in \mathbb{K}^2 \mid \mathcal{P}(\overline{t}) = \mathcal{P}(\overline{\alpha})\}$ can be expressed as

$$\mathcal{F}_{\mathcal{P}}(\overline{\alpha}) = \left\{ \overline{t} \in \mathbb{K}^2 \;\middle|\; H_i^{\mathcal{P}, \overline{\alpha}}(\overline{t}) = 0,\ i = 1, 2, 3 \text{ and } H_4^{\mathcal{P}}(\overline{t}) \neq 0 \right\}$$

(see Lemma 2 in Perez-Diaz and Sendra (2004)).

In these conditions, we introduce the polynomials

$$S_1^{\mathcal{P},\overline{\alpha}}(t_1) = \frac{\text{Content}_Z(\text{Res}_{t_2}(H_1^{\mathcal{P},\overline{\alpha}}, H_2^{\mathcal{P},\overline{\alpha}} + ZH_3^{\mathcal{P},\overline{\alpha}}))}{\text{Content}_{\{Z,W\}}(\text{Res}_{t_2}(H_1^{\mathcal{P},\overline{\alpha}}, H_2^{\mathcal{P},\overline{\alpha}} + ZH_3^{\mathcal{P},\overline{\alpha}} + WH_4^{\mathcal{P}}))},$$

$$S_2^{\mathcal{P},\overline{\alpha}}(t_2) = \frac{\text{Content}_Z(\text{Res}_{t_1}(H_1^{\mathcal{P},\overline{\alpha}}, H_2^{\mathcal{P},\overline{\alpha}} + ZH_3^{\mathcal{P},\overline{\alpha}}))}{\text{Content}_{\{Z,W\}}(\text{Res}_{t_1}(H_1^{\mathcal{P},\overline{\alpha}}, H_2^{\mathcal{P},\overline{\alpha}} + ZH_3^{\mathcal{P},\overline{\alpha}} + WH_4^{\mathcal{P}}))},$$

and one gets the following theorem.

Theorem 5. *There exists a non–empty open subset $\Omega_{\mathcal{P}} \subset \mathbb{K}^2$, such that for every $\overline{\alpha} = (\alpha_1, \alpha_2) \in \Omega_{\mathcal{P}}$ the following statements hold:*

1. *For $i = 1, 2$, the polynomials $S_i^{\mathcal{P},\overline{\alpha}}$ define the t_i–coordinates of the points in $\mathcal{F}_{\mathcal{P}}(\overline{\alpha})$.*
2. *$\mathcal{F}_{\mathcal{P}}(\overline{\alpha}) = \{(A, B) \in \mathbb{K}^2 |\, S_1^{\mathcal{P},\overline{\alpha}}(A) = 0,\ M_A^{\mathcal{P},\overline{\alpha}}(B) = 0\}$, where*

$$M_A^{\mathcal{P},\overline{\alpha}}(t_2) = \frac{\gcd_{\mathbb{F}[t_2]}(H_1^{\mathcal{P},\overline{\alpha}}(A, t_2), H_2^{\mathcal{P},\overline{\alpha}}(A, t_2), H_3^{\mathcal{P},\overline{\alpha}}(A, t_2))}{\gcd_{\mathbb{F}[t_2]}(H_1^{\mathcal{P},\overline{\alpha}}(A, t_2), H_2^{\mathcal{P},\overline{\alpha}}(A, t_2), H_3^{\mathcal{P},\overline{\alpha}}(A, t_2)), H_4^{\mathcal{P}}(A, t_2))}.$$

3. *$\text{Card}(\mathcal{F}_{\mathcal{P}}(\overline{\alpha})) = \deg(\phi_{\mathcal{P}}) = \deg_{t_i}(S_i{}^{\mathcal{P},\overline{\alpha}}(t_i))$. In particular, it holds that $\mathcal{P}$ is proper if and only if, up to constants in $\mathbb{K}^\star$, it holds that $S_i^{\mathcal{P},\overline{\alpha}} = t_i - \alpha_i$.*

3.3 Algorithm and Examples

The results obtained in Subsection 3.2 can be applied to derive two algorithms to compute $\deg(\phi_{\mathcal{P}})$ by avoiding the requirement of computing explicitly the elements in the fibre. One may either compute the degree deterministically using that $\deg(\phi_{\mathcal{P}}) = \deg_{t_i}(S_i{}^{\mathcal{P}}(t_i, \overline{s}))$ (Algorithm Deterministic Degree of Rational Surface Parametrizations) or probabilistically taking into account that $\deg(\phi_{\mathcal{P}}) = \deg_{t_i}(S_i^{\mathcal{P},\overline{\alpha}}(t_i))$ for $\overline{\alpha} \in \Omega_{\mathcal{P}}$ (Algorithm Probabilistic Degree of Rational Surface Parametrizations). The result of this algorithm is correct with probability almost one. These two approaches are given in Figures 2 and 3.

We illustrate the algorithms by two examples. In Example 3, we compute the degree of the rational map induced by the rational surface parametrization introduced in Example 2. For this purpose, we apply Algorithm Deterministic Degree of Rational Surface Parametrizations. In Example 4, we apply Algorithm Probabilistic Degree of Rational Surface Parametrizations, and we also provide the elements of the fibre (see Theorem 5).

Example 3. Let $\mathcal{V}$ be the surface of Example 2 parametrized by

$$\mathcal{P}(t_1, t_2) = \left(\frac{2t_1^2t_2 + t_1^2 + 4t_2 + 3}{2t_2 + 1}, \frac{1}{(t_1^2 + 1)(2t_2 + 1)}, \frac{-(-1 + 2t_1^2t_2 + t_1^2 + 2t_2)}{2t_2 + 1}\right).$$

We apply Algorithm Deterministic Degree of Rational Surface Parametrizations to compute the cardinality of $\mathcal{F}_{\mathcal{P}}(\overline{s})$, i.e. the degree of $\phi_{\mathcal{P}}$. For this purpose, first we determine the polynomials $H_i^{\mathcal{P}}(\overline{t}, \overline{s})$, $i = 1, 2, 3$,

Algorithm Deterministic Degree of Rational Surface Parametrizations.

Given a surface rational parametrization $\mathcal{P}(\overline{t}) = (\frac{p_{11}}{p_{12}}, \frac{p_{21}}{p_{22}}, \frac{p_{31}}{p_{32}})$, in reduced form, the algorithm computes $\deg(\Phi_{\mathcal{P}})$.

1. Check whether the general assumptions introduced in Subsection 3.1 are satisfied, if not, apply a suitable linear change of variables to $\mathcal{P}(\overline{t})$.
2. Compute the polynomials
$$H_j^{\mathcal{P}}(\overline{t}, \overline{s}) = p_{j1}(\overline{t})p_{j2}(\overline{s}) - p_{j2}(\overline{t})p_{j1}(\overline{s}) \in \mathbb{K}[\overline{s}][\overline{t}], \text{ for } j = 1, 2, 3.$$
3. Compute the polynomial
$$S_1^{\mathcal{P}}(t_1, \overline{s}) = \mathrm{pp}_{\overline{s}}(\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}} + ZH_3^{\mathcal{P}})))$$
(similarly if one computes $S_2^{\mathcal{P}}$).
4. Return $\deg_{t_1}(S_1^{\mathcal{P}}(t_1, \overline{s}))$ (see Theorem 4).

Fig. 2. Algorithm Deterministic Degree of Rational Surface Parametrizations

Algorithm Probabilistic Degree of Rational Surface Parametrizations.

Given a surface rational parametrization $\mathcal{P}(\overline{t}) = (\frac{p_{11}}{p_{12}}, \frac{p_{21}}{p_{22}}, \frac{p_{31}}{p_{32}})$, in reduced form, the algorithm computes, with probability almost one, $\deg(\Phi_{\mathcal{P}})$.

1. Check whether the general assumptions introduced in Subsection 3.1 are satisfied, if not, apply a suitable linear change of variables to $\mathcal{P}(\overline{t})$.
2. Take $\overline{\alpha} \in \mathbb{K}^2$, and compute $H_j^{\mathcal{P},\overline{\alpha}}(\overline{t}) = p_{j1}(\overline{t})p_{j2}(\overline{\alpha}) - p_{j2}(\overline{t})p_{j1}(\overline{\alpha})$, for $i = 1, 2, 3$, as well as $H_4^{\mathcal{P}}(\overline{t}) = \mathrm{lcm}(p_{12}, p_{22}, p_{32})$.
3. Compute the polynomial
$$S_1^{\mathcal{P},\overline{\alpha}}(t_1) = \frac{\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P},\overline{\alpha}}, H_2^{\mathcal{P},\overline{\alpha}} + ZH_3^{\mathcal{P},\overline{\alpha}}))}{\mathrm{Content}_{\{Z,W\}}(\mathrm{Res}_{t_2}(H_1^{\mathcal{P},\overline{\alpha}}, H_2^{\mathcal{P},\overline{\alpha}} + ZH_3^{\mathcal{P},\overline{\alpha}} + WH_4^{\mathcal{P}}))},$$
(similarly if one computes $S_2^{\mathcal{P},\overline{\alpha}}$).
4. Return $\deg_{t_1}(S_1{}^{\mathcal{P},\overline{\alpha}}(t_1))$ (see Theorem 5).

Fig. 3. Algorithm Probabilistic Degree of Rational Surface Parametrizations

$$\begin{array}{l} H_1^{\mathcal{P}} = 4s_2s_1^2t_2 + 2s_2s_1^2 + 2t_2s_1^2 + s_1^2 - 2s_2 + 2t_2 - 4t_1^2t_2s_2 - 2t_1^2t_2 - 2s_2t_1^2 - t_1^2, \\ H_2^{\mathcal{P}} = 2t_1^2t_2 + t_1^2 + 2t_2 - 2s_2s_1^2 - s_1^2 - 2s_2, \\ H_3^{\mathcal{P}} = 4t_2 - 4s_2s_1^2t_2 - 2s_2s_1^2 - 2t_2s_1^2 - s_1^2 - 4s_2 + 4t_1^2t_2s_2 + 2t_1^2t_2 + 2s_2t_1^2 + t_1^2, \end{array}.$$

Now, we compute the polynomial $S_1^{\mathcal{P}}(t_1, \overline{s})$. We get

$$S_1^{\mathcal{P}}(t_1, \overline{s}) = \mathrm{pp}_{\overline{s}}(\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}} + ZH_3^{\mathcal{P}}))) = (t_1 - s_1)(t_1 + s_1).$$

Then, we conclude that $\deg(\phi_{\mathcal{P}}) = \deg_{t_1}(S_1^{\mathcal{P}}(t_1, \overline{s})) = 2$ (see Theorem 4).

Example 4. Let $\mathcal{V}$ be the surface parametrized by

$$\mathcal{P}(\overline{t}) = \left(\frac{t_2^2 + t_2 + t_1}{(t_2+t_1)^2 + t_2^2 + t_2 + t_1 + 1}, \frac{1}{(t_2+t_1)^2}, \frac{t_2^2 + t_2 + t_1}{(t_2+t_1)^2} \right).$$

We consider $\overline{\alpha} \in \mathbb{K}^2$, for instance $\overline{\alpha} = (3,2)$, and we apply Algorithm Probabilistic Degree of Rational Surface Parametrizations to compute the cardinality of $\mathcal{F}_{\mathcal{P}}(\overline{\alpha})$. For this purpose, first we determine the polynomials $H_i^{\mathcal{P},\overline{\alpha}}(\overline{t})$, $i = 1, 2, 3$,

$$H_1^{\mathcal{P},\overline{\alpha}}(\overline{t}) = 17t_2^2 + 26t_2 + 26t_1 - 18t_1t_2 - 9t_1^2 - 9,$$

$$H_2^{\mathcal{P},\overline{\alpha}}(\overline{t}) = 25 - t_2^2 - 2t_1t_2 - t_1^2, \quad H_3^{\mathcal{P},\overline{\alpha}}(\overline{t}) = 16t_2^2 + 25t_2 + 25t_1 - 18t_1t_2 - 9t_1^2,$$

as well as the polynomial

$$H_4^{\mathcal{P}}(\overline{t}) = \mathrm{lcm}(p_{12}(\overline{t}), p_{22}(\overline{t}), p_{23}(\overline{t})) = (2t_2^2 + 2t_1t_2 + t_1^2 + t_2 + t_1 + 1)(t_2 + t_1)^2.$$

Now, we compute the polynomial $S_1^{\mathcal{P},\overline{\alpha}}(t_1)$. We get

$$S_1^{\mathcal{P},\overline{\alpha}}(t_1) = \frac{\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P},\overline{\alpha}}(\overline{t}), H_2^{\mathcal{P},\overline{\alpha}}(\overline{t}) + ZH_3^{\mathcal{P},\overline{\alpha}}(\overline{t})))}{\mathrm{Content}_{\{Z,W\}}(\mathrm{Res}_{t_2}(H_1^{\mathcal{P},\overline{\alpha}}(\overline{t}), H_2^{\mathcal{P},\overline{\alpha}}(\overline{t}) + ZH_3^{\mathcal{P},\overline{\alpha}}(\overline{t}) + WH_4(\overline{t})))}$$

$$= \frac{(t_1 - 3)(t_1 - 7)}{1} = (t_1 - 3)(t_1 - 7).$$

We get that $\deg(\phi_{\mathcal{P}}) = \deg_{t_1}(S_1^{\mathcal{P},\overline{\alpha}}(t_1)) = 2$. Let us also compute the elements of the fibre (see Theorem 5). The roots of the polynomial $S_1^{\mathcal{P},\overline{\alpha}}$, i.e. the t_1–coordinates of the elements in the fibre, are $a_1 = 3,\ a_2 = 7$. Now, for every root a_i we compute the polynomial

$$M_{a_i}^{\mathcal{P},\overline{\alpha}}(t_2) = \frac{\gcd_{\mathbb{K}[t_2]}(H_1^{\mathcal{P},\overline{\alpha}}(a_i, t_2), H_2^{\mathcal{P},\overline{\alpha}}(a_i, t_2), H_3^{\mathcal{P},\overline{\alpha}}(a_i, t_2))}{\gcd_{\mathbb{K}[t_2]}(H_1^{\mathcal{P},\overline{\alpha}}(a_i, t_2), H_2^{\mathcal{P},\overline{\alpha}}(a_i, t_2), H_3^{\mathcal{P},\overline{\alpha}}(a_i, t_2), H_4^{\mathcal{P}}(a_i, t_2))}$$

which roots provide the t_2–coordinates of the elements in the fibre. One gets that

$$\mathcal{F}_{\mathcal{P}}(\overline{\alpha}) = \{(3,2), (7,-2)\},$$

which implies that $\deg(\phi_{\mathcal{P}}) = \mathrm{Card}(\mathcal{F}_{\mathcal{P}}(\overline{\alpha})) = 2$.

4 Proper Reparametrization for Surfaces

Let $\mathcal{P}(\overline{t})$ be a rational parametrization of an algebraic surface $\mathcal{V}$, and let $\phi_{\mathcal{P}} : \mathbb{K}^2 \longrightarrow \mathcal{V} \subset \mathbb{K}^3;\ \overline{t} \longmapsto \mathcal{P}(\overline{t})$ be the rational map induced by $\mathcal{P}$. In Section 3, we showed how to determine $\deg(\phi_{\mathcal{P}})$ by means of computations of gcds and

resultants. In addition, if $\deg(\phi_{\mathcal{P}}) = 1$, by applying the results in Section 2, we may compute the inverse of $\phi_{\mathcal{P}}$.

If $\deg(\phi_{\mathcal{P}}) \neq 1$, that is if $\mathcal{P}$ is not proper, Castelnuovo's Theorem states that one may compute a proper rational parametrization of $\mathcal{V}$ from $\mathcal{P}$, and this question is solved *partially* in Perez-Diaz (2006). More precisely, given an algebraically closed field $\mathbb{K}$, and $\mathcal{P}(\overline{t})$ a rational parametrization of a surface $\mathcal{V}$, we decide whether there exists

$$R(\overline{t}) = (r_1(t_1), r_2(t_2)) = \left(\frac{r_{11}(t_1)}{r_{12}(t_1)}, \frac{r_{21}(t_2)}{r_{22}(t_2)}\right) \in (\mathbb{K}(\overline{t}) \setminus \mathbb{K})^2,$$

such that $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$, and $\mathcal{Q}(\overline{t})$ is a proper parametrization of $\mathcal{V}$. In the affirmative case, we compute $R(\overline{t})$, and $\mathcal{Q}(\overline{t})$. We say that $\mathcal{Q}(\overline{t})$ is a proper reparametrization of $\mathcal{P}(\overline{t})$.

The results obtained provide a new and first approach that will be used to solve the important problem of the proper reparametrization for a general parametrized surface. No other result approaching the problem was known up to the moment. In this section, we outline this approach. For this purpose, we follow the notation introduced in Subsection 3.1.

We start with the following theorem where we characterize whether a parametrization $\mathcal{P}(\overline{t})$ of a surface $\mathcal{V}$ can be decomposed as $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$, where $R(\overline{t}) = (r_1(t_1), r_2(t_2)) \in (\mathbb{K}(\overline{t}) \setminus \mathbb{K})^2$, and $\mathcal{Q}(\overline{t})$ is a proper parametrization of $\mathcal{V}$.

Theorem 6. *Let $\mathcal{P}(\overline{t})$ be a rational parametrization of a surface $\mathcal{V}$. The following statements are equivalent:*

1. *$\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$ where $R(\overline{t}) = (r_1(t_1), r_2(t_2)) \in (\mathbb{K}(\overline{t}) \setminus \mathbb{K})^2$, and $\mathcal{Q}(\overline{t})$ is a proper parametrization of $\mathcal{V}$*
2. *$S_1^{\mathcal{P}}(t_1, \overline{s}) = H_1^R(t_1, s_1)^{\ell_2}$, and $S_2^{\mathcal{P}}(t_2, \overline{s}) = H_2^R(t_2, s_2)^{\ell_1}$, where*

$$H_j^R(t_j, s_j) = r_{j1}(t_j) r_{j2}(s_j) - r_{j2}(t_j) r_{j1}(s_j), \quad \gcd(r_{j1}, r_{j2}) = 1,$$

$$r_j(t_j) := r_{j1}(t_j)/r_{j2}(t_j) \in \mathbb{K}[t_j] \setminus \mathbb{K}, \text{ and } \ell_j := \deg_{t_j}(H_j^R), \text{ for } j = 1, 2.$$

Remark 2. One may easily check if the square free parts of the polynomials $S_i^{\mathcal{P}}(t_i, s_i)$, say $M_j(t_j, s_j)$, $j = 1, 2$, are of the form

$$M_j(t_j, s_j) = r_{j1}(t_j) r_{j2}(s_j) - r_{j2}(t_j) r_{j1}(s_j), \qquad \gcd(r_{j1}, r_{j2}) = 1,$$

with $r_j(t_j) := r_{j1}(t_j)/r_{j2}(t_j) \in \mathbb{K}[t_j] \setminus \mathbb{K}$. For this purpose, we write

$$M_j(t_j, s_j) = C_{m_j,j}(t_j) s_j^{m_j} + \cdots + C_{0,j}(t_j), \quad j = 1, 2,$$

such that there exist $C_{\ell_0,j}$ and $C_{n_0,j}$ not being associated polynomials. In these conditions, it holds that there exists polynomials $r_j(t_j) = r_{j1}(t_j)/r_{j2}(t_j) \in \mathbb{K}(t_j), j = 1, 2$ not both constant satisfying that

$$M_j(t_j, s_j) = r_{j1}(t_j)r_{j2}(s_j) - r_2(t_j)r_{j1}(s_j), \qquad \gcd(r_{j1},\, r_{j2}) = 1,$$

if and only if

$$C_{\ell_0,j}(t_j)C_{n_0,j}(s_j) - C_{\ell_0,j}(s_j)C_{n_0,j}(t_j) = c_j M_j(t_j, s_j),\ c_j \in \mathbb{K}^\star,\ j = 1, 2,$$

(see Corollary 1 in Perez-Diaz (2006)).

Let us assume that for $j = 1, 2$, we may write

$$M_j(t_j, s_j) = r_{j1}(t_j)r_{j2}(s_j) - r_2(t_j)r_{j1}(s_j), \qquad \gcd(r_{j1},\, r_{j2}) = 1,$$

with $r_j(t_j) := r_{j1}(t_j)/r_{j2}(t_j) \in \mathbb{K}[t_j] \setminus \mathbb{K}$. Now, we express the polynomials $r_{j1}(t_j)$ and $r_{j2}(t_j)$ as

$$r_{j1}(t_j) = a_{m_j,j}t_j^{m_j} + \cdots + a_{0,j}, \quad r_{j2}(t_j) = b_{m_j,j}t_j^{m_j} + \cdots + b_{0,j}, \quad j = 1, 2$$

with $a_{m_j,j} \neq 0$ or $b_{m_j,j} \neq 0$. Then,

$$M_j(t_j, s_j) = C_{m_j,j}(t_j)s_j^{m_j} + C_{m_j-1}(t_j)s_j^{m_j-1} + \cdots + C_{0,j}(t_j),$$

where $C_{\ell,j}(t) = a_{\ell,j}r_{j2}(t_j) - b_{\ell,j}r_{j1}(t_j)$, for $\ell = 0, \ldots, m_j$, and $j = 1, 2$.
In these conditions, we consider $R(\overline{t}\,) =$

$$\left(\frac{C_{i_0,1}(t_1)}{C_{j_0,1}(t_1)}, \frac{C_{i_1,2}(t_2)}{C_{j_1,2}(t_2)}\right) = \left(\frac{a_{i_0,1}r_{12}(t_1) - b_{i_0,1}r_{11}(t_1)}{a_{j_0,1}r_{12}(t_1) - b_{j_0,1}r_{11}(t_1)}, \frac{a_{i_1,2}r_{22}(t_2) - b_{i_1,2}r_{21}(t_2)}{a_{j_1,2}r_{22}(t_2) - b_{j_1,2}r_{21}(t_2)}\right),$$

where $a_{i_0,1}b_{j_0,1} \neq a_{j_0,1}b_{i_0,1}$, and $a_{i_1,2}b_{j_1,2} \neq a_{i_1,2}b_{j_1,2}$, and we deduce the following theorem.

Theorem 7. *There exists a proper parametrization $\mathcal{Q}(\overline{t}\,)$ of the surface $\mathcal{V}$ satisfying that $\mathcal{P}(\overline{t}\,) = \mathcal{U}(R(\overline{t}\,))$.*

Once the pair of rational functions $R(\overline{t}\,)$ have been computed, one may determine the corresponding proper parametrization by introducing undetermined coefficients and solving a linear system of equations. Alternatively, one may proceed as follows (see Lemma 5 in Perez-Diaz (2006)).

Theorem 8. *Let $\mathcal{P}(\overline{t}\,)$ be a rational parametrization of a surface $\mathcal{V}$ satisfying that $S_1^{\mathcal{P}}(t_1, \overline{s}\,) = H_1^R(t_1, s_1)^{\ell_2}$, and $S_2^{\mathcal{P}}(t_2, \overline{s}\,) = H_2^R(t_2, s_2)^{\ell_1}$, where*

$$H_j^R(t_j, s_j) = r_{j1}(t_j)r_{j2}(s_j) - r_{j2}(t_j)r_{j1}(s_j), \quad \gcd(r_{j1},\, r_{j2}) = 1, \quad j = 1, 2$$

$r_j(t_j) := r_{j1}(t_j)/r_{j2}(t_j) \in \mathbb{K}[t_j] \setminus \mathbb{K}$, and $\ell_j := \deg_{t_j}(H_j^R)$. Let

$$L_i(s_1, t_2, x_i) = \mathrm{Res}_{t_1}(G_i^{\mathcal{P}}(\overline{t}\,, x_i), s_1r_{12}(t_1) - r_{11}(t_1)),$$

and

$$N_i(\overline{s}\,, x_i) = \mathrm{pp}_{x_i}(\mathrm{Res}_{t_2}(L_i(s_1, t_2, x_i), s_2r_{22}(t_2) - r_{21}(t_2))), \qquad i = 1, 2, 3,$$

where pp_{x_i} *denotes the primitive part w.r.t* x_i*. It holds that*

$$N_i(\overline{s}, x_i) = (q_{i2}(\overline{s})x_i - q_{i1}(\overline{s}))^{\deg(r_1)\cdot\deg(r_2)}, \quad i = 1, 2, 3$$

where $q_{i1}(\overline{s})$, $q_{i2}(\overline{s}) \in \mathbb{K}[\overline{s}]$, $\gcd(q_{i1}, q_{i2}) = 1$, *and* $\mathcal{P} = \mathcal{Q}(R)$, *where*

$$\mathcal{Q}(\overline{t}) = \left(\frac{q_{11}(\overline{t})}{q_{12}(\overline{t})}, \frac{q_{21}(\overline{t})}{q_{22}(\overline{t})}, \frac{q_{31}(\overline{t})}{q_{32}(\overline{t})}\right), \quad \textit{and} \quad R(\overline{t}) = (r_1(t_1), r_2(t_2)).$$

4.1 Algorithm of Proper Reparametrization for Surfaces

The results obtained above can be applied to derive an algorithm that decides whether a rational parametrization $\mathcal{P}$ of a surface $\mathcal{V}$, can be expressed as $\mathcal{P} = \mathcal{Q}(R)$, where $R(\overline{t}) = (r_1(t_1), r_2(t_2)) \in (\mathbb{K}(\overline{t}) \setminus \mathbb{K})^2$, and $\mathcal{Q}$ is a proper parametrization of $\mathcal{V}$. In the affirmative case, the algorithm computes R and $\mathcal{Q}$.

The algorithm is given in Figure 4.

We next illustrate Algorithm Proper Reparametrization for Surfaces with two examples. In Example 5, we consider the parametrization introduced in Example 2. In Example 2, we proved that $\mathcal{P}$ was not proper, and in fact in Example 3, we showed that $\deg(\phi_{\mathcal{P}}) = 2$. In Example 5, we compute a proper parametrization from $\mathcal{P}$.

Example 5. Let $\mathcal{V}$ be the rational surface of Example 3 defined by the parametrization

$$\mathcal{P}(t_1, t_2) = \left(\frac{2t_1^2t_2 + t_1^2 + 4t_2 + 3}{2t_2 + 1}, \frac{1}{(t_1^2 + 1)(2t_2 + 1)}, \frac{-(-1 + 2t_1^2t_2 + t_1^2 + 2t_2)}{2t_2 + 1}\right).$$

In Step 1 of the algorithm, we compute the polynomials $H_i^{\mathcal{P}}(\overline{t}, \overline{s})$, for $i = 1, 2, 3$ (see Example 3). Now, we determine $S_j^{\mathcal{P}}(t_j, \overline{s})$. We obtain

$$S_1^{\mathcal{P}}(t_1, s_1) = (t_1 - s_1)(t_1 + s_1), \qquad S_2^{\mathcal{P}}(t_2, s_2) = (t_2 - s_2)^2.$$

Since $\deg_{t_j}(S_j^{\mathcal{P}}) > 1$, and $S_j^{\mathcal{P}}(t_j, \overline{s}) \in \mathbb{K}[t_j, s_j]$ for $j = 1, 2$, we go to Step 5 of the algorithm, and we determine the polynomials

$$M_1(t_1, s_1) = (t_1 - s_1)(t_1 + s_1) = C_{2,1}(t_1)s_1^2 + C_{0,1}(t_1),$$

where $C_{2,1} = -1$, $C_{0,1} = t_1^2$, and

$$M_2(t_2, s_2) = (t_2 - s_2) = C_{1,2}(t_2)s_2 + C_{0,2}(t_2),$$

where $C_{1,2} = -1$, $C_{0,2} = t_2$. Now, we consider

$$R(\overline{t}) = (C_{0,1}(t_1)/C_{2,1}(t_1), C_{0,2}(t_2)/C_{1,2}(t_2)) = (-t_1^2, -t_2).$$

In Step 5.2 we check that

Algorithm Proper Reparametrization for Surfaces.

Given a rational affine parametrization $\mathcal{P} = (p_{11}/p_{12},\, p_{21}/p_{22},\, p_{31}/p_{32})$, in reduced form, of an algebraic surface $\mathcal{V}$, the algorithm decides whether $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$, with $R(\overline{t}) = (r_1(t_1), r_2(t_2)) \in \mathbb{K}(\overline{t})^2$, and in the affirmative case, it computes $R(\overline{t})$ and the rational proper parametrization $\mathcal{Q}(\overline{t})$.

1. Compute $H_j^{\mathcal{P}}(\overline{t}, \overline{s}) = p_{j1}(\overline{t})p_{j2}(\overline{s}) - p_{j2}(\overline{t})p_{j1}(\overline{s}), \quad j = 1, 2, 3.$
2. Determine the polynomials

$$S_1^{\mathcal{P}} = \mathrm{pp}_{\overline{s}}(\mathrm{Content}_Z(\mathrm{Res}_{t_2}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}} + ZH_3^{\mathcal{P}}))),$$

$$S_2^{\mathcal{P}} = \mathrm{pp}_{\overline{s}}(\mathrm{Content}_Z(\mathrm{Res}_{t_1}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}} + ZH_3^{\mathcal{P}})))$$

3. If $\deg_{t_j}(S_j^{\mathcal{P}}) = 1$ return $\mathcal{Q}(\overline{t}) = \mathcal{P}(\overline{t})$ and $R(\overline{t}) = (t_1, t_2)$. Otherwise go to Step 4.
4. If $S_j^{\mathcal{P}}(t_j, \overline{s}) \in \mathbb{K}[t_j, s_j]$, for $j = 1, 2$, go to Step 5. Otherwise, return "$\mathcal{P}$ cannot be expressed as $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$, with $R(\overline{t}) = (r_1(t_1), r_2(t_2))$".
5. Compute the polynomial

$$M_j(t_j, s_j) = \mathrm{Sqrfree}(S_j^{\mathcal{P}}(t_j, \overline{s})) = C_{m_j,j}(t_j)s_j^{m_j} + \cdots + C_{0,j}(t_j), \quad j = 1, 2,$$

where $\mathrm{Sqrfree}(p)$ denotes the square free part of a polynomial p.

5.1. Consider $R(\overline{t}) = (r_1(t_1), r_2(t_2))$ with

$$r_j(t_j) = r_{j1}(t_j)/r_{j2}(t_j) = C_{\ell_0,j}(t_j)/C_{n_0,j}(t_j) \in \mathbb{K}(t_j), \quad j = 1, 2$$

and such that $C_{\ell_0,j}$ and $C_{n_0,j}$ are not associated polynomials.

5.2. If $C_{\ell_0,j}(t_j)C_{n_0,j}(s_j) - C_{\ell_0,j}(s_j)C_{n_0,j}(t_j) = c_j M_j(t_j, s_j)$, $c_j \in \mathbb{K}^\star$, $j = 1, 2$, then go to Step 6. Otherwise, return "$\mathcal{P}$ cannot be expressed as $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$, with $R(\overline{t}) = (r_1(t_1), r_2(t_2))$".

6. For $i = 1, 2, 3$, compute the polynomials

$$L_i(s_1, t_2, x_i) = \mathrm{Res}_{t_1}(G_i^{\mathcal{P}}(\overline{t}, x_i), s_1 r_{12}(t_1) - r_{11}(t_1)),$$

$$N_i(\overline{s}, x_i) = \mathrm{pp}_{x_i}(\mathrm{Res}_{t_2}(L_i(s_1, t_2, x_i), s_2 r_{22}(t_2) - r_{21}(t_2))) =$$
$$(q_{i2}(\overline{s})x_i - q_{i1}(\overline{s}))^{\deg(r_1)\deg(r_2)},$$

where $G_i^{\mathcal{P}}(\overline{t}, x_i) = x_i p_{i2}(\overline{t}) - p_{i1}(\overline{t})$.

7. Return

$$\mathcal{Q}(\overline{t}) = \left(q_{11}(\overline{t})/q_{12}(\overline{t}),\, q_{21}(\overline{t})/q_{22}(\overline{t}),\, q_{31}(\overline{t})/q_{32}(\overline{t})\right),$$

and $R(\overline{t}) = (C_{\ell_0,1}(t_1)/C_{n_0,1}(t_1),\, C_{\ell_0,2}(t_2)/C_{n_0,2}(t_2))$.

Fig. 4. Algorithm Proper Reparametrization for Surfaces

$$M_1 = C_{2,1}(t_1)C_{0,1}(s_1) - C_{2,1}(s_1)C_{0,1}(t_1),$$
$$M_2 = C_{1,2}(t_2)C_{0,2}(s_2) - C_{1,2}(s_2)C_{0,2}(t_2).$$

Note that $C_{0,i}$, $C_{2,i}$ are not associated polynomials for $i = 1, 2$. In Step 6 of the algorithm, we compute the polynomials

$$L_i(s_1, x_i) = \mathrm{Res}_{t_1}(G_i^{\mathcal{P}}(\overline{t}, x_i), s_1 r_{12}(t_1) - r_{11}(t_1)),$$

$$N_i(\overline{s}, x_i) = \mathrm{pp}_{x_i}(\mathrm{Res}_{t_2}(L_i(s_1, x_i), s_2 r_{22}(t_2) - r_{21}(t_2))) =$$
$$(q_{i2}(\overline{s})x_i - q_{i1}(\overline{s}))^{\deg_{t_1}(r_1)\cdot \deg_{t_2}(r_2)},$$

where $G_i^{\mathcal{P}}(\overline{t}, x_i) = x_i p_{i2}(\overline{t}) - p_{i1}(\overline{t})$. We get

$$N_1(\overline{s}, x_1) = (x_1 - 3 - s_1 + 2s_2x_1 - 4s_2 - 2s_1s_2)^2,$$
$$N_2(\overline{s}, x_2) = (-1 + x_2 + s_1x_2 + 2s_2x_2 + 2s_2s_1x_2)^2,$$
$$N_3(\overline{s}, x_3) = (x_3 - 1 + s_1 + 2s_2x_3 + 2s_2 + 2s_1s_2)^2.$$

Finally, in Step 7, the algorithm returns the proper parametrization

$$\mathcal{Q}(\overline{t}) = \left(\frac{3 + t_1 + 4t_2 + 2t_1t_2}{2t_2 + 1}, \frac{1}{1 + t_1 + 2t_2 + 2t_1t_2}, -\frac{2t_2 - 1 + t_1 + 2t_1t_2}{2t_2 + 1}\right).$$

and $R(\overline{t}) = (-t_1^2, -t_2)$.

Example 6. Let $\mathcal{V}$ be the rational surface defined by the parametrization

$$\mathcal{P}(\overline{t}) = \Bigg(\frac{{t_1}^4{t_2}^4 + 2\,{t_1}^4{t_2}^2 + 5\,{t_1}^4 + 2\,{t_2}^4 + 4\,{t_2}^2 + 11}{{t_2}^4 + 2\,{t_2}^2 + 5},$$

$$\frac{6 + {t_1}^4{t_2}^4 + 2\,{t_1}^4{t_2}^2 + 5\,{t_1}^4 + {t_2}^4 + 2\,{t_2}^2}{({t_2}^4 + 2\,{t_2}^2 + 5)\,({t_1}^4 + 1)}, \frac{3 + {t_1}^4{t_2}^4 + 2\,{t_1}^4{t_2}^2 + 5\,{t_1}^4 + {t_2}^4 + 2\,{t_2}^2}{-({t_2}^4 + 2\,{t_2}^2 + 5)}\Bigg).$$

In Step 1 of the algorithm, we compute the polynomials

$H_1^{\mathcal{P}}(\overline{t}, \overline{s}) = -2t_1^4t_2^4s_2^2 - 25t_1^4 - t_1^4t_2^4s_2^4 + t_2^4 + 2t_2^2 + 25s_1^4 - s_2^4 - 2s_2^2 - 2t_1^4t_2^2s_2^4 + 2s_1^4s_2^2t_2^4 + 4s_1^4s_2^2t_2^2 + 2s_1^4s_2^4t_2^2 + s_1^4s_2^4t_2^4 - 4t_1^4t_2^2s_2^2 + 10s_1^4t_2^2 - 5t_1^4t_2^4 - 10t_1^4t_2^2 + 5s_1^4s_2^4 + 5s_1^4t_2^4 + 10s_1^4s_2^2 - 5t_1^4s_2^4 - 10t_1^4s_2^2,$

$H_2^{\mathcal{P}}(\overline{t}, \overline{s}) = 5t_1^4 + t_2^4 + 2t_2^2 - 5s_1^4 - s_2^4 - 2s_2^2 + t_1^4t_2^4 + 2t_1^4t_2^2 - s_1^4s_2^4 - 2s_1^4s_2^2,$

$H_3^{\mathcal{P}}(\overline{t}, \overline{s}) = 2t_1^4t_2^4s_2^2 + 25t_1^4 + t_1^4t_2^4s_2^4 + 2t_2^4 + 4t_2^2 - 25s_1^4 - 2s_2^4 - 4s_2^2 + 2t_1^4t_2^2s_2^4 - 2s_1^4s_2^2t_2^4 - 4s_1^4s_2^2t_2^2 - 2s_1^4s_2^4t_2^2 - s_1^4s_2^4t_2^4 + 4t_1^4t_2^2s_2^2 - 10s_1^4t_2^2 + 5t_1^4t_2^4 + 10t_1^4t_2^2 - 5s_1^4s_2^4 - 5s_1^4t_2^4 - 10s_1^4s_2^2 + 5t_1^4s_2^4 + 10t_1^4s_2^2.$

Now, we determine $S_j^{\mathcal{P}}(t_j, \overline{s})$. We obtain

$$S_1^{\mathcal{P}}(t_1, s_1) = (t_1 - s_1)^4(t_1 + s_1)^4(t_1^2 + s_1^2)^4,$$

$$S_2^{\mathcal{P}}(t_2, s_2) = (t_2 + s_2)^4 (t_2 - s_2)^4 (t_2^2 + 2 + s_2^2)^4.$$

Since $\deg_{t_j}(S_j^{\mathcal{P}}) > 1$, and $S_j^{\mathcal{P}}(t_j, \overline{s}) \in \mathbb{K}[t_j, s_j]$ for $j = 1, 2$, we go to Step 5 of the algorithm, and we determine the polynomials

$$M_1(t_1, s_1) = (t_1 - s_1)(t_1 + s_1)(t_1^2 + s_1^2) = C_{4,1}(t_1) s_1^4 + C_{0,1}(t_1),$$

where $C_{4,1} = -1$, $C_{0,1} = t_1^4$, and

$$M_2(t_2, s_2) = (t_2 + s_2)(t_2 - s_2)(t_2^2 + 2 + s_2^2) = C_{4,2}(t_2) s_2^4 + C_{2,2}(t_2) s_2^2 + C_{0,2}(t_2),$$

where $C_{4,2} = -1$, $C_{2,2} = -2$, $C_{0,2} = t_2^4 + 2t_2^2$. Now, we consider

$$R(\overline{t}) = (C_{0,1}(t_1)/C_{4,1}(t_1), C_{0,2}(t_2)/C_{4,2}(t_2)) = (-t_1^4, -t_2^2(2 + t_2^2)).$$

In Step 5.2 we check that

$$M_1 = C_{4,1}(t_1) C_{0,1}(s_1) - C_{4,1}(s_1) C_{0,1}(t_1)$$

$$M_2 = C_{4,2}(t_2) C_{0,2}(s_2) - C_{4,2}(s_2) C_{0,2}(t_2).$$

Note that $C_{0,i}$, $C_{4,i}$ are not associated polynomials for $i = 1, 2$. In Step 6 of the algorithm, we compute the polynomials

$$L_i(s_1, x_i) = \mathrm{Res}_{t_1}(G_i^{\mathcal{P}}(\overline{t}, x_i), s_1 r_{12}(t_1) - r_{11}(t_1)),$$

$$N_i(\overline{s}, x_i) = \mathrm{pp}_{x_i}(\mathrm{Res}_{t_2}(L_i(s_1, x_i), s_2 r_{22}(t_2) - r_{21}(t_2))) =$$

$$(q_{i2}(\overline{s}) x_i - q_{i1}(\overline{s}))^{\deg_{t_1}(r_1) \cdot \deg_{t_2}(r_2)},$$

where $G_i^{\mathcal{P}}(\overline{t}, x_i) = x_i p_{i2}(\overline{t}) - p_{i1}(\overline{t})$. We get

$$N_1(\overline{s}, x_1) = (11 - 5x_1 - 5s_1 + s_2 x_1 - 2s_2 + s_2 s_1)^{16},$$

$$N_2(\overline{s}, x_2) = (-6 + 5x_2 - 5s_1 x_2 + 5s_1 - s_2 x_2 + s_2 + s_2 s_1 x_2 - s_2 s_1)^{16},$$

$$N_3(\overline{s}, x_3) = (-s_2 x_3 - s_2 + s_2 s_1 - 5s_1 + 5x_3 + 3)^{16}.$$

Finally, in Step 7, the algorithm returns the proper parametrization

$$\mathcal{Q}(\overline{t}) = \left(-\frac{11 + t_2 t_1 - 5t_1 - 2t_2}{t_2 - 5}, \frac{6 - t_2 - 5t_1 + t_2 t_1}{5 - 5t_1 - t_2 + t_2 t_1}, \frac{-t_2 + t_2 t_1 - 5t_1 + 3}{t_2 - 5} \right),$$

and $R(\overline{t}) = (t_1^4, t_2^2(2 + t_2^2))$.

Acknowledgement. The author was partially supported by the Spanish Ministerio de Educación y Ciencia under Project MTM2005-08690-C02-01.

References

Arrondo, E., Sendra, J., Sendra, J.R.: Parametric Generalized Offsets to Hypersurfaces. Journal of Simbolic Computation. 23, 267–285 (1997)

Alonso, C., Gutierrez, J., Recio, T.: A Rational Function Decomposition Algorithm by Near-separated Polynomials. Journal of Symbolic Computation. 19, 527–544 (1995)

Chionh, E.W., Goldman, R.N.: Degree, Multiplicity and Inversion Formulas for Rational Surfaces using u-Resultants. Computer Aided Geometric Design. 9/2, 93–109 (1992)

Cox, D.A., Sederberg, T.W., Chen, F.: The Moving Line Ideal Basis of Planar Rational Curves. Computer Aided Geometric Design. 8, 803–827 (1998)

Gutierrez, J., Rubio, R., Sevilla, D.: On Multivariate Rational Decomposition. Journal of Symbolic Computation. 33, 545–562 (2002)

Harris, J.: Algebraic Geometry. A First Course. Springer, Heidelberg (1995)

Hoffmann, C.M., Sendra, J.R., Winkler, F.: Parametric Algebraic Curves and Applications. Journal of Symbolic Computation. 23 (1997)

Hoschek, J., Lasser, D.: Fundamentals of Computer Aided Geometric Design. A.K. Peters Wellesley MA. Ltd (1993)

Pérez-Díaz, S., Schicho, J., Sendra, J.R.: Properness and Inversion of Rational Parametrizations of Surfaces. Applicable Algebra in Engineering, Communication and Computing 13, 29–51 (2002)

Pérez-Díaz, S., Sendra, J.R.: Computation of the Degree of Rational Surface Parametrizations. Journal of Pure and Applied Algebra. 193/1-3, 99–121 (2004)

Pérez-Díaz, S., Sendra, J.R.: Partial Degree Fomulae for Rational Algebraic Surfaces. In: Proc. ISSAC-2005, pp. 301–308. ACM Press, New York (2005)

Pérez-Díaz, S.: On the Problem of Proper Reparametrization for Rational Curves and Surfaces. Computer Aided Geometric Design. 23/4, 307–323 (2006)

Schicho, J.: Inversion of Birational Maps with Gröbner Basis. In: Buchberger, B., Winkler, F. (eds.) Unobstructed Shortest Paths in Polyhedral Environments. Lectures Notes Series, vol. 251, pp. 495–503. Cambridge Univ. Press, Cambridge (1998)

Sederberg, T.W.: Improperly Parametrized Rational Curves. Computer Aided Geometric Design. 3, 67–75 (1986)

Sendra, J.R., Winkler, F.: Tracing Index of Rational Curve Parametrizations. Computer Aided Geometric Design. 18/8, 771–795 (2001)

Shafarevich, I.R.: Basic Algebraic Geometry Schemes; 1 Varieties in Projective Space, vol. 1. Springer, Berlin, New York (1994)

Discrete Surfaces in Isotropic Geometry

Helmut Pottmann[1] and Yang Liu[2]

[1] Geometric Modeling and Industrial Geometry, Vienna University of Technology, Wiedner Hauptstr. 8-10, A-1040 Wien, Austria
pottmann@geometrie.tuwien.ac.at
[2] Department of Computer Science, University of Hong Kong, China
yliu@cs.hku.hk
http://www.geometrie.tuwien.ac.at/ig

Abstract. Meshes with planar quadrilateral faces are desirable discrete surface representations for architecture. The present paper introduces new classes of planar quad meshes, which discretize principal curvature lines of surfaces in so-called isotropic 3-space. Like their Euclidean counterparts, these isotropic principal meshes meshes are visually expressing fundamental shape characteristics and they can satisfy the aesthetical requirements in architecture. The close relation between isotropic geometry and Euclidean Laguerre geometry provides a link between the new types of meshes and the known classes of conical meshes and edge offset meshes. The latter discretize Euclidean principal curvature lines and have recently been realized as particularly suited for freeform structures in architecture, since they allow for a supporting beam layout with optimal node properties. We also present a discrete isotropic curvature theory which applies to all types of meshes including triangle meshes. The results are illustrated by discrete isotropic minimal surfaces and meshes computed by a combination of optimization and subdivision.

Keywords: discrete differential geometry, surfaces in architecture, isotropic geometry, conical mesh, edge offset mesh, isotropic minimal surface.

1 Introduction

Recently, *discrete differential geometry* enjoys increasing interest in Computer Graphics and Geometric Modeling. This emerging field at the border between differential and discrete geometry provides attractive tools for processing discrete surface representations with methods that may be seen as an extension of the classical theory. This is quite different from numerical differential geometry where approximation by smooth representations is directly or indirectly used as a preprocessing step in order to employ the classical results.

Very recently it turned out that the precise geometric results and relations offered by discrete differential geometry are of high importance in architectural design [7,10,11,13,19,21]. This is rooted in the fact that in architectural freeform structures which are based on polyhedral surfaces the mesh representation (i)

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 341–363, 2007.
© Springer-Verlag Berlin Heidelberg 2007

is seen and greatly influences the aesthetics and (ii) is the basis for the actual construction.

Though meshes are very well understood in geometric computing, their application to architecture is difficult for at least two reasons. One is that triangle meshes, though easy to compute and inherently stable due to their geometry, lead to complications in node manufacturing and supporting beam layout [19,21]. The other reason is that alternatives, namely quad meshes or hexagonal meshes with planar faces, are harder to deal with: Quad meshes with planar faces discretize so-called conjugate curve networks [24]; their computation in general requires nonlinear optimization [13] and aesthetical requirements may be very hard to achieve. Planarity of faces in hexagonal meshes is easy to realize since only three faces meet in a vertex, but an aesthetic layout of such a mesh on a given freeform surface is an unsolved problem. Building on recently achieved progress [7,13,19,21], the present paper introduces new remarkable classes of planar quad meshes. These are formulated and studied with help of so-called isotropic geometry, which is motivated as follows.

1. Principal curvature lines are sometimes viewed as appropriate curves for guiding the force flow and for visually expressing fundamental shape characteristics. These are reasons why they are preferred for beam layout. The ideal force flow depends on the direction of gravity, whereas Euclidean principal curvature lines are independent of it. Isotropic geometry has a distinguished (isotropic) direction (cf. Section 2.1) and thus it is promising to investigate whether isotropic principal curvature lines provide structural advantages. Before doing so, we need tools to design meshes whose polygons are aligned in isotropic principal direction. This is a main goal of the present paper. An example is shown in Fig. 1. It also illustrates that isotropic principal curvature lines may lead to aesthetically pleasing meshes. Isotropic principal curvature lines form a special conjugate curve network, namely the one which appears as orthogonal network in the top view (projection in isotropic direction).
2. Conical meshes [7,13] and edge offset meshes [7,21] have various advantages for architecture, including a supporting beam layout with optimized nodes [7,13,21]. These two types of meshes are objects of Laguerre geometry. Isotropic geometry occurs naturally as a model of Laguerre geometry. Hence, it is interesting to see how these meshes appear in the so-called isotropic model of Laguerre geometry. We have here a principle of transfer between meshes. This is useful both for the theoretical understanding and for applications.

1.1 Previous Work

Discrete differential geometry. Particularly important in the present context are results on quadrilateral meshes with planar faces (*PQ meshes*). PQ meshes discretize so-called conjugate curve networks on surfaces [24]. They are a basic concept in the integrable system viewpoint of discrete differential geometry [4]. *Circular meshes* [14] and *conical meshes* [13] are special PQ meshes which discretize the network of principal curvature lines. In a circular mesh, each face

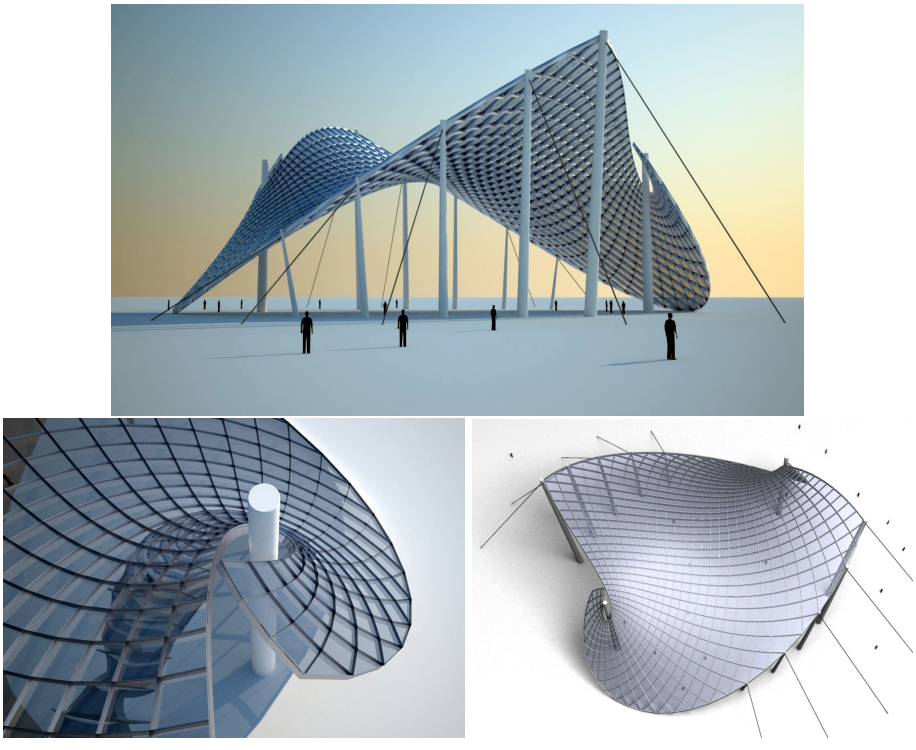


Fig. 1. This architectural design study is based on a quadrilateral mesh with planar faces which discretizes the network of isotropic principal curvature lines on a minimal surface of isotropic geometry

has a circum-circle; in a conical mesh, the face planes meeting at a vertex are tangent to a cone of revolution. Principal curvature lines are a concept of Lie sphere geometry, and thus circular and conical meshes may be treated in a unified way within Lie geometry [5]. Elementary relations between circular and conical meshes are discussed in [20]. *Mesh parallelism* turned out to be an important tool for studying polyhedral surfaces with a view towards applications in architecture [7,21]. Due to its high importance for the present paper, mesh parallelism is briefly reviewed in subsection 3.1. PQ meshes which are discrete counterparts to minimal surfaces are the topic of several contributions (e.g. [2,3,21,33]).

Geometry processing. Computational issues concerning the meshes mentioned above are only addressed in a few papers [2,13,21]. PQ meshes, in particular circular and conical meshes, may be designed by a combination of subdivision and nonlinear optimization [13]. The same optimization algorithm can be used to approximate a given shape by a circular/conical mesh, if the input mesh is aligned along a network of principal curvature lines. Methods for the design and computation of meshes with offset properties relevant for architecture and for

the layout of supporting beams with optimized nodes are based on the concept of parallel meshes [21].

Approximation of smooth surfaces by meshes with planar faces, without a focus on beam layout and offset properties, can be performed via variational shape approximation [9]. Cutler and Whiting [10] modified this method with regard to aesthetics and architectural design. Research on geometry for architecture in general is promoted by the Smart Geometry group (*www.smartgeometry.com*).

Laguerre geometry is the geometry of oriented planes and spheres in Euclidean 3-space [1,8]. It has been applied in the study of rational curves and surfaces with rational offsets [16,18] and in the reconstruction of developable surfaces from point clouds [15]. Its recent application in the study of conical meshes [5,13,33] and meshes with edge offsets [21] is the most important one for the present paper and will be continued in Section 3.

Isotropic geometry is based on a simple semi-Riemannian metric [22]; cf. Section 2.1. It naturally appears when properties of functions are to be geometrically visualized and interpreted at hand of their graph surfaces [17]. In particular, this holds for the visualization of stress properties in planar elastic systems at hand of their Airy surfaces [30]. An application of isotropic geometry in Image Processing has been given by Koenderink and van Doorn [12].

1.2 Contributions and Overview

The contributions of the present paper are the following:

1. We provide an introduction into isotropic geometry, Laguerre geometry and the relations between them (Section 2).
2. Meshes which discretize the network of isotropic principal curvature lines are studied in Section 3. There, we also address optimization algorithms for their design and computation.
3. We introduce discrete isotropic curvatures for meshes with planar faces and provide examples for discrete minimal surfaces of isotropic geometry (Sec. 4).

2 Fundamentals

2.1 Isotropic Geometry

Motions and metric. Isotropic geometry has been developed by Strubecker in the 1940s. The results we need for the present investigations may be found in [25] or in the monograph by Sachs [22]. Isotropic 3-space I^3 is based on the following group G_6 of affine transformations $(x, y, z) \mapsto (x', y', z')$ in $\mathbb{R}^3$,

$$\begin{aligned} x' &= a + x\cos\phi - y\sin\phi, \\ y' &= b + x\sin\phi + y\cos\phi, \\ z' &= c + c_1 x + c_2 y + z, \quad a, b, c, c_1, c_2, \phi \in \mathbb{R}, \end{aligned} \tag{1}$$

which are called *isotropic congruence transformations (i-motions)*. We see that i-motions appear as Euclidean motions (translation vector $(a, b)^T$ and rotation

angle ϕ) in the projection onto the xy-plane; the result of this projection $\mathbf{p} = (x, y, z) \mapsto \mathbf{p}' = (x, y, 0)$ is called *top view* henceforth. Hence, an isotropic congruence transformation is composed of a Euclidean motion in the xy-plane and an affine shear transformation in z-direction. Many metric properties in isotropic 3-space I^3 (invariants under G_6) are actually Euclidean invariants in the top view. For example, one defines the *i-distance* of two points $\mathbf{x}_j = (x_j, y_j, z_j),\ j = 1, 2$, as the Euclidean distance of their top views $\mathbf{x}'_j$,

$$\|\mathbf{x}_1 - \mathbf{x}_2\|_i := \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{2}$$

Thus, two points (x, y, z_j) with the same top view (called *parallel points*) have i-distance zero, but they need not agree. Since the i-metric (2) degenerates along z-parallel lines, these lines are called *isotropic lines*. Due to the architectural applications we have in mind, the isotropic z-direction is to be thought as vertical. *Isotropic angles* between straight lines are measured as Euclidean angles in the top view.

Planes, circles and spheres. There are two types of planes in I^3.

(i) *Non-isotropic planes* are not parallel to the z-direction. In these planes we basically have a Euclidean metric: This is not the one we are used to, since we we have to make the usual Euclidean measurements in the top view. An *i-circle (of elliptic type)* in a non-isotropic plane P is an ellipse, whose top view is a Euclidean circle. Such an i-circle with center $\mathbf{m} \in P$ and radius r is the set of all points $\mathbf{x} \in P$ with $\|\mathbf{x} - \mathbf{m}\|_i = r$.

(ii) *Isotropic planes* are parallel to the z-axis. There, I^3 induces an isotropic metric. An *isotropic circle (of parabolic type)* is a parabola with z-parallel axis and thus it lies in an isotropic plane. An i-circle of parabolic type is not the iso-distance set of a fixed point.

There are also two types of *isotropic spheres*. An *i-sphere S of the cylindrical type* is the set of all points $\mathbf{x} \in I^3$ with $\|\mathbf{x} - \mathbf{m}\|_i = r$. Speaking in a Euclidean way, such a sphere is a right circular cylinder with z-parallel rulings; its top view is the Euclidean circle with center $\mathbf{m}'$ and radius r. Any point which is parallel to $\mathbf{m}$ lies on the axis of this cylinder and may also serve as center of S. The more interesting and important type of spheres are the *i-spheres of parabolic type*,

$$z = \frac{A}{2}(x^2 + y^2) + Bx + Cy + D,\ A \neq 0. \tag{3}$$

From a Euclidean perspective, they are paraboloids of revolution with z-parallel axis. The intersections of these i-spheres with planes P are i-circles: If P is not isotropic, then the intersection is an i-circle of elliptic type. If P is isotropic, the intersection curve is an i-circle of parabolic type.

Curvature theory of surfaces. In the present context, isotropic differential geometry of surfaces is of great importance. To avoid degeneracies, we study only those surfaces Φ which are regular and do not possess isotropic (i.e., vertical) tangent planes. Thus, we may write Φ in explicit form,

$$\Phi:\ z = f(x, y). \tag{4}$$

The isotropic version of Gaussian curvature theory uses the isotropic unit sphere,

$$\Sigma : z = \frac{1}{2}(x^2 + y^2) =: s(x, y). \tag{5}$$

To each point $\mathbf{x} = (x, y, f(x, y)) \in \Phi$ we associate as Gaussian image that point $\sigma(\mathbf{x}) = \tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, s(\tilde{x}, \tilde{y})) \in \Sigma$ whose tangent plane is parallel to the tangent plane of Φ at $\mathbf{x}$. This requires agreement of the gradients $\nabla f(x, y) = (f_x, f_y)$ and $\nabla s(\tilde{x}, \tilde{y}) = (\tilde{x}, \tilde{y})$ and thus yields for the Gaussian mapping σ,

$$\sigma : \ (x, y, f(x, y)) \mapsto (f_x, f_y, \frac{1}{2}(f_x^2 + f_y^2)). \tag{6}$$

The derivative of σ is the *isotropic shape operator*. It is a linear mapping between the parallel and thus identified tangent spaces of Φ at $\mathbf{x}$ and Σ at $\tilde{\mathbf{x}}$. The top view of the shape operator acts on vectors $(t_1, t_2)^T \in \mathbb{R}^2$ and is given by

$$(t_1, t_2)^T \mapsto \nabla^2 f \cdot (t_1, t_2)^T. \tag{7}$$

Its transformation matrix is the Hessian $\nabla^2 f$ of f,

$$\nabla^2 f = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}.$$

The eigenvectors of the shape operator determine the isotropic principal curvature directions. They are conjugate directions and appear as orthogonal vectors in the top view (eigenvectors of $\nabla^2 f$); the corresponding eigenvalues are called *i-principal curvatures* κ_1, κ_2. Their product equals $\det(\nabla^2 f)$ and is called *isotropic curvature* (or *relative curvature*)

$$K = \kappa_1 \kappa_2 = f_{xx} f_{yy} - f_{xy}^2. \tag{8}$$

Isotropic mean curvature H is given by the Laplacian of f,

$$2H = \kappa_1 + \kappa_2 = \operatorname{trace}(\nabla^2 f) = f_{xx} + f_{yy} = \Delta f. \tag{9}$$

Isotropic minimal surfaces are characterized by $H = 0$ and thus they are graphs of harmonic functions f ($\Delta f = 0$). They may have branching points (see e.g. Fig. 8) and thus the surfaces can have several sheets; we understand a 'graph' in this general sense. Isotropic minimal surfaces possess many properties which are analogous to their Euclidean counterparts [22,31]. We will construct discrete i-minimal surfaces in Section 4.

Finally we point to the integral curves of the field of principal directions. These are called *isotropic principal curvature lines*. They constitute exactly that conjugate curve network on S which appears as orthogonal network in the top view. Discrete counterparts form the content of Section 3.

Metric duality. In projective 3-space we have a principle of duality. To give some examples, points are dual to planes and vice versa, straight lines are dual to straight lines and inclusions are reversed. However, duality cannot be applied to

metric quantities of Euclidean geometry. This is different in isotropic geometry, which enjoys a metric duality. It may be realized by the polarity with respect to the i-sphere (5), which maps point $\mathbf{p} = (p_1, p_2, p_3)$ to plane $P: \ z = p_1x+p_2y-p_3$. Points $\mathbf{p}$ and $\mathbf{q} = (q_1, q_2, q_3)$ with i-distance d (from (2), recall $d^2 = (p_1 - q_1)^2 + (p_2 - q_2)^2$) are mapped to planes P and $Q: \ z = q_1x + q_2y - q_3$. The i-angle ϕ of the two planes P, Q equals d (we may even define i-angles in this way) and thus ϕ is simply computed as $\phi^2 = (p_1 - q_1)^2 + (p_2 - q_2)^2$.

Parallel points, i.e. points with the same top view, in the duality correspond to parallel planes. A surface $\Phi: z = f(x, y)$, seen as set of contact elements (points plus tangent planes) corresponds to a surface Φ^*, parameterized by

$$x^* = f_x(x, y), \ y^* = f_y(x, y), \ z^* = xf_x + yf_y - f. \tag{10}$$

Contact elements along i-principal curvature lines of Φ and Φ^* correspond in the duality. Note that Φ^* may have singularities which correspond to parabolic surface points of Φ ($K = 0$). This is reflected in the following relations between the isotropic curvature measures of dual surface pairs [32],

$$H^* = H/K, \ K^* = 1/K. \tag{11}$$

Thus, the dual surface to an i-minimal surface is also i-minimal. For further properties of the metric duality, see [22]. We will use duality in Section 4 to obtain curvatures at mesh vertices from curvatures attached to faces. In particular, duality will allow us to derive curvatures in triangle meshes from those in hexagonal meshes.

Already this very brief introduction into isotropic geometry reveals that it is actually simpler than Euclidean geometry. Isotropic counterparts to nonlinear problems of Euclidean geometry may be linear and thus computationally less demanding than their Euclidean version. For example, equation (9) shows that the PDEs characterizing minimal surfaces or surfaces of constant mean curvature are linear in isotropic geometry; the corresponding equations of Euclidean geometry are nonlinear. Since we use these tools mainly for aesthetical shape generation, it may be sufficient to resort to the isotropic version.

In order to present the relation between isotropic geometry and Laguerre geometry, we have to discuss sphere transformations.

Isotropic sphere transformations. In I^3, there exists a counterpart to Euclidean Möbius geometry (recall that planes and spheres form the set of so-called Möbius spheres and Möbius transformations act bijectively on this set). In I^3, one puts *i*-spheres of parabolic type and non-isotropic planes into the same class $\mathcal{S}$ of *isotropic Möbius spheres*; they are given by (3), including $A = 0$. Together with an appropriate extension $\mathcal{P}$ of the point set of I^3 (one adds $\mathbb{R}$ as set of 'ideal points'), there exists a group M^i_{10} of so-called *isotropic Möbius (i-M) transformations* which act bijectively in $\mathcal{P}$ and in $\mathcal{S}$. The group of i-M-transformations also acts bijectively on the set of *i-M-circles*. An i-M-circle is defined as intersection of two i-M-spheres and may be an i-circle of elliptic or parabolic type or a non-isotropic line. The 10-dimensional group M^i_{10} is isomorphic to the group of Euclidean Laguerre transformations (see subsection 2.2).

The top view of an i-M-transformation is a planar Euclidean Möbius transformation. The basic i-M-transformations are inversions with respect to i-spheres. The inversion (reflection) at an i-M-sphere $S : z = A(x^2 + y^2)/2 + ... =: s(x, y)$ is given by $(x, y, z) \mapsto (x, y, 2s(x, y) - z)$. The top view remains unchanged and in z-direction we have a reflection at the corresponding point of S. An inversion κ with respect to an i-sphere S of cylindrical type appears in the top view as ordinary inversion with respect to the circle S', the top view of S. Also in 3D, corresponding points $\mathbf{x}, \kappa(\mathbf{x})$ lie collinear with a fixed center $\mathbf{c}$ which must be contained in the axis of S.

2.2 Laguerre Geometry

Laguerre geometry is the geometry of oriented planes and oriented spheres in Euclidean E^3 [1,8]. We may write an or. plane P in Hesse normal form $\mathbf{n}^T \cdot \mathbf{x} + h = 0$, where the unit normal vector $\mathbf{n}$ defines the orientation; $\mathbf{n}^T \cdot \mathbf{x} + h$ is the signed distance of the point $\mathbf{x}$ to P. An oriented sphere S, with center $\mathbf{m}$ and signed radius r, is tangent to an oriented plane P if the signed distance of $\mathbf{m}$ to P equals r, i.e., $\mathbf{n}^T \cdot \mathbf{m} + h = r$. Points are viewed as or. spheres with radius zero.

A *Laguerre transformation (L-transformation)* is a mapping which is bijective on the sets of or. planes and or. spheres, respectively, and keeps plane/sphere tangency. Viewing or. spheres S as points $\mathbf{S} := (\mathbf{m}, r) \in \mathbb{R}^4$, an L-transformation is seen in $\mathbb{R}^4$ as a special affine map $\mathbf{S}' = \mathbf{a} + R \cdot \mathbf{S}$, where R describes a linear map which preserves the inner product $\langle \mathbf{x}, \mathbf{y} \rangle := x_1y_1 + x_2y_2 + x_3y_3 - x_4y_4$. With the diagonal matrix $D := diag(1, 1, 1, -1)$ we have $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \cdot D \cdot \mathbf{y}$, and the condition on R reads $R^T \cdot D \cdot R = D$. A simple example of an L-transformation is the offsetting operation (given by the identity matrix R and $\mathbf{a} = (0, 0, 0, d)$), which adds a constant d to the radius of each sphere and in particular maps a point to a sphere of radius d.

Let us return to the standard model of Laguerre geometry in E^3. A pencil of parallel or. planes has the same normal vector $\mathbf{n}$ (image point on the 'Gaussian' sphere S^2). An L-transformation keeps the parallelity of or. planes and induces a Möbius transformation of the Gaussian sphere S^2. The or. planes which are tangent to two or. spheres envelope an or. right circular cone, and thus or. cones (including limit cases) are also objects of Laguerre geometry.

The isotropic model of Laguerre geometry. There is the following remarkable relation between Laguerre geometry and isotropic Möbius geometry. We may use $(\mathbf{n}, h)$ as coordinates of an or. plane P. However, these four coordinates are not independent due to $\|\mathbf{n}\| = 1$. Thus, one replaces $\mathbf{n} = (n_1, n_2, n_3)$ (point of S^2) by its image point $(n_1/(n_3 + 1), n_2/(n_3 + 1), 0)$ under the stereographic projection of S^2 from $(0, 0, -1)$ onto $z = 0$. It is then convenient (and can be explained in a more geometric way, see e.g. [18]) to view the point

$$P^i := \frac{1}{n_3 + 1}(n_1, n_2, h) \tag{12}$$

as a kind of dual image of the oriented plane $P : n_1x + n_2y + n_3z + h = 0$. P^i should be seen as a point of isotropic space I^3, whose point set has been extended to $\mathcal{P}$

(see above). Thus, one also speaks of the *isotropic model of Laguerre geometry*. Parallel or. planes P, Q appear in the isotropic model as parallel points P^i, Q^i. The or. tangent planes of an or. sphere S are seen as points of an isotropic Möbius sphere in the isotropic model. The common tangent planes of two or. spheres (= or. cone) correspond to the common points of two i-M-spheres (= i-M-circle). A non-developable surface Φ viewed as set of or. tangent planes is mapped to a surface Φ^i in the isotropic model. Tangent planes along a Euclidean principal curvature line of Φ are mapped to points of an isotropic principal curvature line of Φ^i. Underlying all these facts is the essential result that an L-transformation corresponds to an i-M-transformation in the isotropic model. Hence, the two groups are isomorphic.

These classical relations will be important to realize the close connection between certain known meshes of Euclidean geometry and the meshes of isotropic geometry which form the content of the present paper. For more results and details on Laguerre geometry, see [1,8,18].

3 Principal Meshes in Isotropic Geometry

3.1 Meshes with Planar Faces

Here, we first introduce some essential methods and results on mesh parallelism [7,21] and then apply them in the context of isotropic geometry.

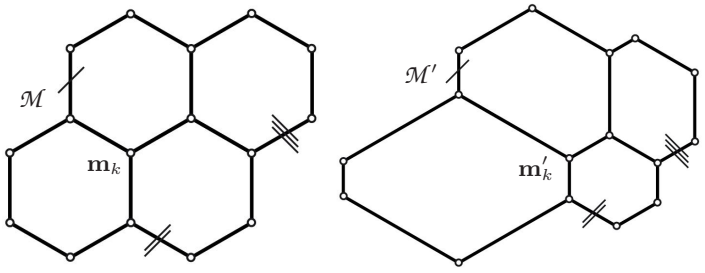


Fig. 2. Meshes $\mathcal{M}, \mathcal{M}'$ with planar faces are parallel if they are combinatorially equivalent and corresponding edges are parallel

Mesh parallelism. A mesh $\mathcal{M}$ is represented by its vertices, concatenated in $(\mathbf{m}_1, \ldots, \mathbf{m}_N) \in \mathbb{R}^{3N}$ and the combinatorics, i.e., edges and faces. If $\mathcal{M}', \mathcal{M}''$ have the same combinatorics, a linear combination $\lambda'\mathcal{M}' + \lambda''\mathcal{M}''$ is defined vertex-wise; this operation corresponds to the linear combination of vectors in $\mathbb{R}^{3N}$. Meshes $\mathcal{M}, \mathcal{M}'$ are *parallel*, if they have the same combinatorics and corresponding edges are parallel (see Fig. 2). We use this definition only if the *faces of $\mathcal{M}$ (and hence of $\mathcal{M}'$) are planar*. Clearly, corresponding faces of $\mathcal{M}$ and $\mathcal{M}'$ lie in parallel planes. The set of meshes parallel to $\mathcal{M}$ is denoted by $\mathcal{P}(\mathcal{M})$. This space does not only contain 'nice' meshes; we may see various undesirable effects

such as unevenly distributed face sizes, sharp regression edges or overlapping regions in meshes of this space, but it is theoretically and practically important to use it. Even visually unpleasant meshes of $\mathcal{P}(\mathcal{M})$ are helpful in the computation of optimized supporting beam layouts (cf. [21]).

Since triangles with parallel edges are scaled copies of each other, two parallel triangle meshes are scaled copies of each other. This degeneracy does not matter since we do not consider triangle meshes anyway (except at the end of Sec. 4 where a different type of parallelism is used).

Suppose $\mathcal{M}', \mathcal{M}'' \in \mathcal{P}(\mathcal{M})$. Then, for each edge $\mathbf{m}_i\mathbf{m}_j$, the vectors $\mathbf{m}_i' - \mathbf{m}_j'$, $\mathbf{m}_i'' - \mathbf{m}_j''$ are multiples of $\mathbf{m}_i - \mathbf{m}_j$. It follows that any expression $(\lambda'\mathbf{m}_i' + \lambda''\mathbf{m}_i'') - (\lambda'\mathbf{m}_j' + \lambda''\mathbf{m}_j'')$ is a multiple of $\mathbf{m}_i - \mathbf{m}_j$. This shows that the linear combination $\lambda'\mathcal{M}' + \lambda''\mathcal{M}''$ is also parallel to $\mathcal{M}$, and thus $\mathcal{P}(\mathcal{M})$ is a linear subspace of $\mathbb{R}^{3N}$. The zero vector of $\mathcal{P}(\mathcal{M})$ is the mesh $\mathbf{o} = 0 \cdot \mathcal{M}$, all of whose vertices coincide with the origin of the coordinate system.

Meshes with offsets. A mesh $\mathcal{M}' \in \mathcal{P}(\mathcal{M})$ at constant distance from $\mathcal{M}$ is called an *offset* of $\mathcal{M}$. Different ways to define the precise meaning of a constant offset distance, $\operatorname{dist}(\mathcal{M}, \mathcal{M}') = d$, lead to different kinds of offsets:

(i) *vertex offsets*: $\|\mathbf{m}_i - \mathbf{m}_i'\| = d$, independently of the vertex $\mathbf{m}_i$.
(ii) *edge offsets*: The distance of corresponding parallel edges (actually, lines which carry those edges) does not depend on the edge and equals d.
(iii) *face offsets*: The distance of faces (actually, planes which carry faces) is independent of the face and equals d.

To characterize offset pairs of parallel meshes $\mathcal{M}, \mathcal{M}'$, one defines a further parallel mesh, the *Gauss image* $\mathcal{S} = (\mathcal{M}' - \mathcal{M})/d$, which satisfies $\operatorname{dist}(\mathcal{S}, \mathbf{o}) = 1$. Then, the following result is easy to show [21]:

(i) For the vertex offsets, the vertices of the Gauss image $\mathcal{S}$ are contained in the unit sphere S^2. If $\mathcal{S}$ is a quad mesh and no edges degenerate, then $\mathcal{M}$ has a vertex offset if and only if it is *circular*, i.e., each face has a circum-circle.
(ii) For edge offsets, the edges of the Gauss image $\mathcal{S}$ are tangent to S^2. These *edge offset meshes* are studied in [21] and allow for a realization as architectural designs with beams of constant height and the cleanest possible nodes.
(iii) For the face offsets, the faces of the Gauss image $\mathcal{S}$ are tangent to S^2. A mesh has a face offset if and only if it is *conical*, i.e., the faces around each vertex are tangent to a cone of revolution.

Hence, if $\mathcal{M}$ has the v-offset (e-offset, f-offset, resp.) property, any parallel mesh including $\mathcal{S}$ has the same offset property.

Relative Gauss mapping and r-principal meshes. The condition that parallel meshes $\mathcal{M}, \mathcal{M}'$ are at constant distance (in the notation from above, $\operatorname{dist}(\mathcal{M}', \mathcal{M}) = d$) is too rigid for certain applications in architecture, in particular for the layout of fair planar quad meshes, see [21]. One therefore considers the following generalization: We let $\mathcal{M}' = \mathcal{M} + d\mathcal{S}$, where $\mathcal{S} \in \mathcal{P}(\mathcal{M})$ is a mesh which approximates a certain convex surface Σ, then called the 'relative unit sphere'.

The mesh $\mathcal{S}$, which is parallel to $\mathcal{M}$, is called *r-Gauss image* of $\mathcal{M}$. The vector $\mathbf{s}_i$ of $\mathcal{S}$ corresponding to a vertex $\mathbf{m}_i$ of $\mathcal{M}$ is the *r-normal vector*. Then the r-offset at r-distance d has vertices $\mathbf{m}_i + d\mathbf{s}_i$. The correspondence $\sigma : \mathcal{M} \to \mathcal{S}$ between the meshes $\mathcal{M}$ and $\mathcal{S}$ is called *r-Gauss mapping*. We use here the terminology from *relative differential geometry* [28], which generalizes Euclidean differential geometry of curves and surfaces by choosing a different 'unit sphere' Σ instead of the standard Euclidean one. Below, we will choose a paraboloid Σ.

Definition 1. *If $\mathcal{M}$ and $\mathcal{S}$ are parallel quad meshes such that $\mathcal{S}$ approximates the r-unit sphere Σ, we call $\mathcal{M}$ an* r-principal mesh.

Here, 'approximation' needs to be specified. We have seen three important examples above: $\mathcal{S}$ may have its vertices on Σ ($\mathcal{S}$ inscribed to Σ), it may have its face planes tangent to Σ ($\mathcal{S}$ circumscribed to Σ) or its edges may be tangent to Σ (here, $\mathcal{S}$ is called 'midscribed' to Σ [27]). An r-principal mesh can be considered as discrete counterpart to the network of relative principal curvature lines of a smooth surface. One reason for this is the following: The relative normals $\mathbf{v}_i + t\mathbf{s}_i$, $\mathbf{s}_i \in \mathcal{S}$, $\mathbf{v}_i \in \mathcal{M}$, along each mesh row or column of $\mathcal{M}$ form a discrete model of a developable surface (since normals at consecutive vertices are coplanar; see [21]).

Mesh parallelism in isotropic space. Without explicitly mentioning it, we have described isotropic surface theory in terms of relative differential geometry with the paraboloid Σ of (5) as r-unit sphere. The presented concepts are ideally suited to study principal meshes in I^3 (*i-principal meshes*). Before doing this, we address some consequences of metric duality in I^3 onto mesh parallelism and principal meshes in I^3.

Consider a mesh $\mathcal{M}$ and a parallel mesh $\mathcal{S}$ approximating the isotropic sphere Σ. Corresponding *face planes are parallel* and thus we call $\mathcal{M}$ and $\mathcal{S}$ *f-parallel*.

Let us now apply metric duality, namely the polarity with respect to Σ, to an f-parallel mesh pair $(\mathcal{M}, \mathcal{S})$. Corresponding vertices $\mathbf{m}_i^*, \mathbf{s}_i^*$ of the dual meshes $\mathcal{M}^*, \mathcal{S}^*$ are the poles of corresponding (parallel) face planes M_i, S_i of $\mathcal{M}, \mathcal{S}$. Thus, corresponding vertices $\mathbf{m}_i^*, \mathbf{s}_i^*$ are parallel points, i.e., their top views agree; we call $\mathcal{M}^*, \mathcal{S}^*$ *v-parallel*. The v-parallel pair $(\mathcal{M}^*, \mathcal{S}^*)$ has identical top views and the combinatorics is dual to that of $(\mathcal{M}, \mathcal{S})$. However, corresponding face planes of $(\mathcal{M}^*, \mathcal{S}^*)$ are no longer parallel. Clearly, the meshes which are v-parallel to $\mathcal{M}^*$ form a linear space.

Both $\mathcal{S}$ and $\mathcal{S}^*$ approximate Σ; if one mesh is inscribed, the other is circumscribed, whereas being midscribed remains unchanged under duality. By Def. 1 we can say: *If $\mathcal{M}$ and $\mathcal{S}$ are f-parallel quad meshes and $\mathcal{S}$ approximates the i-unit sphere Σ, then $\mathcal{M}$ is an i-principal mesh.*

In the smooth setting, metric duality preserves i-principal curvature lines. Thus we can also use the dual approach: *An i-principal mesh $\mathcal{M}$ is v-parallel to a mesh $\mathcal{S}$ approximating Σ*. Now viewing I^3 as isotropic model of Laguerre geometry, we may perform the change to the standard model in E^3. This is a geometric transformation τ which maps $\mathcal{M} \subset I^3$ to a mesh $\mathcal{N} = \tau(\mathcal{M}) \subset E^3$. Corresponding parallel vertices of $\mathcal{M}$ and $\mathcal{S}$ are transformed into corresponding parallel face planes of $\mathcal{N}$ and $\mathcal{T} = \tau(\mathcal{S})$. Since $\mathcal{T}$ is an approximation of the

Euclidean sphere $\tau(\Sigma)$ and f-parallel to $\mathcal{N}$, the mesh $\mathcal{N}$ is a Euclidean principal mesh. More precise versions of this general result will be given below.

3.2 Conical and Circular Meshes in Isotropic Geometry

In this section we describe two main classes of principal meshes in I^3 and derive properties which are fundamental for their design and computation. These meshes contain aesthetically pleasing ones, which will be demonstrated in section 3.3. We even expect structural advantages, which is a topic of ongoing research.

An *i-conical mesh* $\mathcal{M} \in I^3$ is parallel to a quad mesh $\mathcal{S}$ whose planar faces are tangent to an isotropic sphere Σ of parabolic type such as (5). All tangent planes of Σ which pass through a vertex $\mathbf{s}_i \in \mathcal{S}$ envelope an *isotropic cone of revolution* Δ_i and thus the face planes of $\mathcal{S}$ which meet at $\mathbf{s}_i$ are tangent to Δ_i. By parallelity of corresponding faces in $\mathcal{S}$ and $\mathcal{M}$ also the faces of $\mathcal{M}$ which meet at the corresponding vertex $\mathbf{v}_i \in \mathcal{M}$ are tangent to an isotropic cone of revolution Γ_i. The cone Γ_i arises from Δ_i by the translation $\mathbf{s}_i \mapsto \mathbf{v}_i$.

Let us now apply duality to the f-parallel pair $(\mathcal{M}, \mathcal{S})$. As discussed above, we obtain a v-parallel pair $(\mathcal{M}^*, \mathcal{S}^*)$, where $\mathcal{S}^*$ has its vertices $\mathbf{s}_i^*$ on Σ; $\mathbf{s}_i^*$ are the points of tangency between the face planes of $\mathcal{S}$ and Σ. Hence, any quad Q^* in $\mathcal{S}^*$ has an isotropic circum-circle (intersection of Q^*'s plane and Σ). This means that the top view of each face Q^*, which agrees with the top view of the corresponding face of $\mathcal{M}^*$, has a Euclidean circum-circle. Thus also all faces in $\mathcal{M}^*$ have an isotropic circum-circle and we have to call $\mathcal{M}^* \in I^3$ an *i-circular mesh*. We see that in I^3 conical and circular meshes are dual to each other. Recall that regularity of surfaces is not preserved under duality. Thus duality is practically only useful if $\mathcal{S}$ is regular (without over-foldings), which implies that $\mathcal{M}$ and $\mathcal{M}^*$ approximate surfaces without parabolic surface points.

Let us now show that any i-circular mesh $\mathcal{M}$ is not only v-parallel to an inscribed mesh $\mathcal{S}$ of Σ, but also f-parallel to another inscribed mesh $\mathcal{S}_1$ of Σ. We pick an arbitrary vertex $\mathbf{v}_i \in \mathcal{M}$ and define a point $\mathbf{s}_i \in \Sigma$ as corresponding vertex of $\mathcal{S}_1$. By parallelism of corresponding edges we can now uniquely construct $\mathcal{S}_1$. As in Euclidean geometry, there are infinitely many parallel meshes $\mathcal{S}_1$ of $\mathcal{M}$ which are inscribed to Σ. Conversely, any mesh $\mathcal{M}$ which is parallel to an inscribed mesh $\mathcal{S}_1$ of Σ is circular: If we denote the isotropic angles at the vertices of a quad $Q \in \mathcal{S}_1$ by $\alpha_1, \ldots, \alpha_4$ (these are the Euclidean angles in the top view, Fig. 3, left), being circular means that the sum of opposite angles equals π,

$$\alpha_1 + \alpha_3 = \alpha_2 + \alpha_4 = \pi. \tag{13}$$

Since the angles of the quad $F \in \mathcal{M}$ parallel to Q are the same, also F has an isotropic circum-circle and thus $\mathcal{M}$ is i-circular. Let us summarize our results:

Theorem 1. *I-circular meshes discretize the network of isotropic principal curvature lines. An i-circular mesh $\mathcal{M}$ is v-parallel to an inscribed PQ mesh $\mathcal{S}$ of the i-unit sphere Σ. Moreover, it is f-parallel to infinitely many inscribed PQ meshes $\mathcal{S}_1$ of Σ. Each quad in $\mathcal{M}$, $\mathcal{S}$ and $\mathcal{S}_1$ has an isotropic circum-circle, i.e., the top views of these meshes are Euclidean circle patterns (Fig. 3, left).*

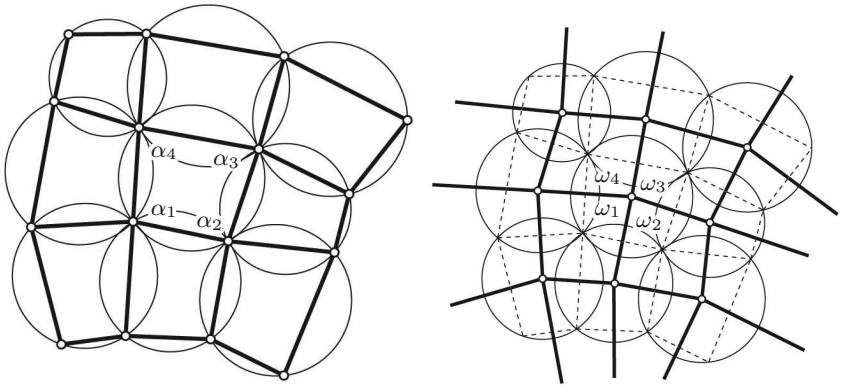


Fig. 3. The top view of an i-circular mesh is a Euclidean circle pattern (left). The centers of the circles define the dual mesh, which appears as top view of an i-conical mesh (right).

By duality between i-circular and i-conical meshes and between inscribed and circumscribed meshes of Σ we see that any conical mesh M is v-parallel to infinitely many circumscribed meshes $\mathcal{S}_1$ of Σ. Moreover, we note that the pole of a plane Q with respect to Σ has as top view the center of the circle which appears as top view of the i-circle $Q \cap \Sigma$. This proves the following result:

Theorem 2. *I-conical meshes discretize the network of isotropic principal curvature lines. They are dual to i-circular meshes. An i-conical mesh $\mathcal{M}$ is f-parallel to a circumscribed mesh $\mathcal{S}$ of the i-unit sphere Σ. It is also v-parallel to infinitely many circumscribed meshes $\mathcal{S}_1$ of Σ. The top view $\mathcal{M}'$ of $\mathcal{M}$ is a quad mesh, whose vertices are the centers of circles in a pattern and thus $\mathcal{M}'$ is a special Voronoi diagram (Fig. 3, right).*

By duality, an i-conical mesh has at each vertex the angle balance condition

$$\omega_1 + \omega_3 = \omega_2 + \omega_4 = \pi. \tag{14}$$

These are isotropic angles and may be measured as Euclidean angles in the top view (cf. Fig. 3, right). (14) also follows from the fact that the vertices of $\mathcal{M}'$ are centers of circles in a pattern.

Remark 1. I-circular meshes are closely related to conical meshes of Euclidean geometry. The change τ from the isotropic model of Laguerre geometry to the standard model maps an i-circular mesh $\mathcal{M}$ to a conical mesh $\mathcal{N} = \tau(\mathcal{M})$ in E^3. This follows immediately from the fact that the i-circum-circles of $\mathcal{M}$'s faces are mapped to the tangent cones of the face planes meeting at $\mathcal{N}$'s vertices. Any isotropic Möbius transformation maps an i-circular mesh $\mathcal{M}$ (seen as set of vertices) to another i-circular mesh $\mathcal{M}_1$ (Fig. 4). The transformation under τ yields the known invariance of Euclidean conical meshes (viewed as sets of oriented face planes) under Laguerre transformations.

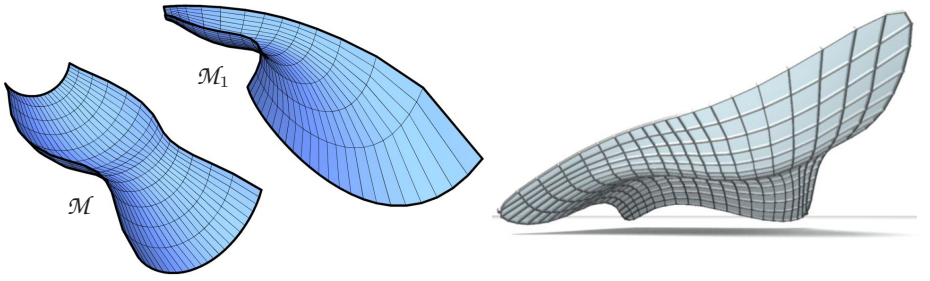


Fig. 4. The i-circular mesh $\mathcal{M}$ (left) has been obtained by combined Catmull-Clark subdivision and optimization. The image $\mathcal{M}_1$ of $\mathcal{M}$ under an isotropic Möbius transformation is again an i-circular mesh. The mesh on the right hand side has been obtained by subdivision and optimization towards an i-conical mesh.

3.3 Optimization Algorithms

The computation of i-circular and i-conical meshes can be performed with algorithms from Liu et al. [13]. We just have to use conditions (13) and (14), respectively, as constraints. A simple way for designing such meshes is to alternate between subdivision and optimization (Figs. 4 and 5). It should be noted that optimization, even if this seems to be the case if we read the description of algorithms above, does not magically convert any given mesh into an i-circular or i-conical mesh: Only initial meshes which do not deviate too much from one with the desired properties will behave in a controlled and expected way. Since it is easier to define a coarse mesh which has nearly planar faces and further desired properties (e.g. being close to a discretely i-orthogonal mesh), the combination with subdivision usually works quite well. Although planarity of faces and other constraints are in general destroyed by any linear subdivision method, this can be 'repaired' in the optimization phase.

3.4 Isothermic Meshes and Their Dual Counterparts in I^3

A further remarkable type of i-principal meshes $\mathcal{M}$ are those which are v-parallel to a planar quad mesh $\mathcal{S}$ all of whose edges touch the i-sphere Σ. Each face plane of $\mathcal{S}$ intersects Σ in an isotropic circle which is tangent to the edges of that face. If we transform Σ into a Euclidean sphere with an appropriate projective map , $\mathcal{S}$ is mapped to a Koebe polyhedron [34]. Thus, we may call $\mathcal{S}$, which is midscribed to Σ, an *isotropic Koebe polyhedron*. The top view of the isotropic Koebe polyhedron is a quad mesh with the following properties: (i) each face has an inscribed circle, and (ii) per edge there is only one point of tangency with the inscribed circles of the adjacent faces. This shows the existence of an orthogonal circle packing (Fig. 6). The arising orthogonal circle patterns have been first discussed by Schramm [26]; their computation from given combinatorics amounts to the minimization of a convex function [6]. We call $\mathcal{M}$ an *i-isothermic mesh*; it is an isotropic counterpart of the S-isothermic meshes by Bobenko at al. [2]. As

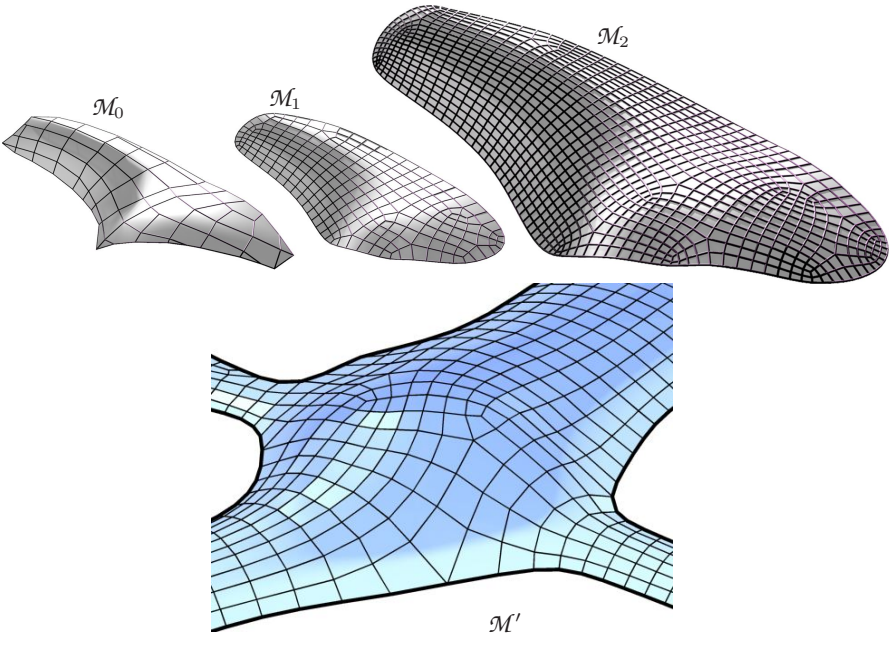


Fig. 5. Meshes $\mathcal{M}_0$, $\mathcal{M}_1$ and $\mathcal{M}_2$ (enlarged) have been generated by repeated subdivision and i-conical optimization. The same holds for the i-conical mesh $\mathcal{M}'$, but here the side conditions that $\mathcal{M}'$ approximates a given design plus its boundary resulted in a loss of smoothness.

in the Euclidean case, *these meshes are not suitable for approximating arbitrary surfaces.* They discretize isothermic surfaces of I^3. In section 4 we will investigate a particularly beautiful class of i-isothermic meshes, namely discrete versions of i-minimal surfaces.

Theorem 3. *I-isothermic meshes discretize the network of isotropic principal curvature lines on isothermic surfaces in I^3. An i-isothermic mesh is v-parallel to an isotropic Koebe polyhedron and has a top view mesh whose faces have inscribed circles which belong to a Schramm circle pattern (Fig. 6).*

Remark 2. If we view an i-isothermic mesh $\mathcal{M}$ as set of i-circles inscribed to its faces, an i-Möbius transformation maps $\mathcal{M}$ to another i-isothermic mesh. Moreover, if we map the set of inscribed i-circles via the transfer τ to the standard model of Laguerre geometry in E^3, we obtain a set of right circular cones Γ_j, whose vertices $\mathbf{g}_j$ form a quad mesh $\mathcal{N}$ with planar faces and the edge offset property (for direct constructions of the meshes $\mathcal{N}$, see [21]). The cone Γ_j with vertex $\mathbf{g}_j$ contains the 4 edges emanating from $\mathbf{v}_j$. The invariance of i-isothermicity under i-Möbius transformations proves the invariance of edge offset quad meshes $\mathcal{N}$ under Laguerre transformations if $\mathcal{N}$ is viewed as set of vertex cones Γ_j.

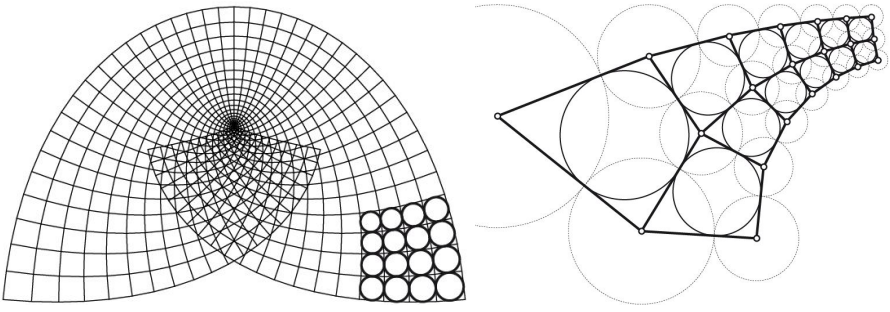


Fig. 6. The top view of an i-isothermic mesh is a quad mesh whose faces have inscribed circles. These circles are part of a circle packing (left). Quadruples of points of tangency are co-circular; these orthogonal circles form a second circle packing (right).

4 Isotropic Curvatures in Meshes with Planar Faces

The aesthetics of meshes is essential for artistic applications such as architecture. Aesthetics is closely related to the curvature behavior and thus it is important to have an adapted discrete curvature theory at our disposal. This is the topic of the present section. We will illustrate the results with discrete minimal surfaces, i.e., surfaces with vanishing discrete i-mean curvature.

We consider a mesh $\mathcal{M} \in I^3$ with planar faces, not necessarily a quad mesh, and an f-parallel mesh $\mathcal{S}$ which approximates Σ. Thus $\mathcal{S}$ may be viewed as *i-Gauss image* $\sigma(\mathcal{M})$ of $\mathcal{M}$. For such a situation, an adapted discrete curvature theory has been developed recently [21]. We briefly describe this theory, using the simplifications which arise in I^3, and later present a dual counterpart which is more specific to isotropic geometry.

I-curvatures at faces. Let F be a face of $\mathcal{M}$ with vertices $\mathbf{v}_0, \dots, \mathbf{v}_{k-1}$. The corresponding parallel face $S = \sigma(F)$ of $\mathcal{S}$ shall have vertices $\mathbf{s}_0, \dots, \mathbf{s}_{k-1}$. The isotropic area of a domain can be measured as usual area in the top view. Let $\mathbf{v}'_i, \mathbf{s}'_i \in \mathbb{R}^2$ denote the top views of the involved vertices. The signed areas of the top views of F and S (isotropic areas of F and S) can be computed as

$$\text{area}(F') = \frac{1}{2}\sum_{j=0}^{k-1} \det(\mathbf{v}'_j, \mathbf{v}'_{j+1}),\ \ \text{area}(S') = \frac{1}{2}\sum_{j=0}^{k-1} \det(\mathbf{s}'_j, \mathbf{s}'_{j+1}), \tag{15}$$

with indices modulo k. We also need the so-called *mixed area* of the two polygons F' and S', which is a well known concept in convex geometry [**?**],

$$\text{area}(F', S') := \frac{1}{4}\sum_{j=0}^{k-1} [\det(\mathbf{v}'_j, \mathbf{s}'_{j+1}) + \det(\mathbf{s}'_j, \mathbf{v}'_{j+1})]. \tag{16}$$

I-curvature K and i-mean curvature H of mesh $\mathcal{M}$ at face F are then defined as

$$K(F) = \frac{\text{area}(S')}{\text{area}(F')}, \quad H(F) = \frac{\text{area}(F', S')}{\text{area}(F')}. \tag{17}$$

It has been shown in [21] that these definitions of curvatures in many aspects behave as their counterparts in the smooth setting. An example is the following: If F, S or any linear combination of them is convex, one can prove $H(F)^2 - K(F) \geq 0$ and thus compute principal curvatures $\kappa_1(F), \kappa_2(F)$ which satisfy $H(F) = [\kappa_1(F) + \kappa_2(F)]/2$ and $K(F) = \kappa_1(F)\kappa_2(F)$. A further example is the curvature behavior of the offset meshes $\mathcal{M}^d = \mathcal{M} + d\mathcal{S}$,

$$K(F^d) = \frac{K(F)}{1 + 2dH(F) + d^2K(F)}, \; H(F^d) = \frac{H(F) + dK(F)}{1 + 2dH(F) + d^2K(F)}. \tag{18}$$

This includes the following discrete isotropic version of a known result on surfaces of constant mean curvature in Euclidean geometry: If $\mathcal{M}$ has constant isotropic mean curvature $H \neq 0$, then its offset at distance $d = -1/H$ has constant i-mean curvature $-H$ and the offset at distance $d = -1/(2H)$ has constant i-Gaussian curvature $K = 4H^2$.

Remark 3. We have adopted common sign conventions in isotropic geometry which differ from the Euclidean ones at some places. This is seen in the last remark on offsets, in the sign of H, and in the fact that we defined the derivative (not the negative derivative) of the i-Gauss mapping as shape operator.

I-minimal surfaces and Christoffel duality. Discrete minimal surfaces in I^3 are meshes with $H = 0$, i.e. with

$$\text{area}(F, S) = 0, \tag{19}$$

for all parallel face pairs (F, S), which is equivalent to $\text{area}(F', S') = 0$. According to (9), they discretize graph surfaces of harmonic functions. A simple characterization of parallel quads $F = 1234$ and $S = 1^*2^*3^*4^*$ with vanishing mixed area has been presented in [21],

$$\text{area}(F, S) = 0 \iff 13 \text{ parallel } 2^*4^*, \; 24 \text{ parallel } 1^*3^*. \tag{20}$$

This is illustrated in Fig. 7, left. Note that the two quads differ in their orientation. If F has an inscribed circle, so does S (Fig. 7, left). It may seem easy now to construct isotropic minimal surfaces. However, starting from a mesh $\mathcal{S}$ which approximates Σ, it will in general not be possible to derive a minimal mesh $\mathcal{M}$ by a face-wise construction from the faces $S \in \mathcal{S}$, since the construction of the four faces around a vertex does not close. If the construction is possible in a consistent way, one calls the pair $(\mathcal{M}, \mathcal{S})$ a *Christoffel-dual pair* (Fig. 7, right). Christoffel duality should not be mixed up with metric duality.

We have already presented meshes to which Christoffel duality can be applied. These are the i-isothermic meshes, in particular isotropic Koebe meshes. *The Christoffel dual of an isotropic Koebe mesh is a discrete i-minimal surface, and*

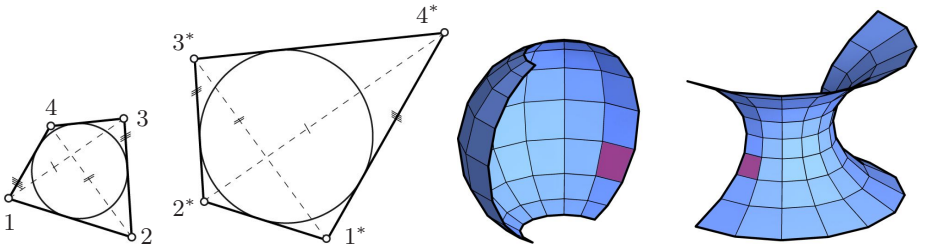


Fig. 7. Left: Two parallel quads have vanishing mixed area if they possess opposite orientation and parallel diagonals. If one of the two quads possesses an inscribed circle, so does the other. Right: Christoffel dual meshes are parallel meshes where all pairs of corresponding faces have vanishing mixed area.

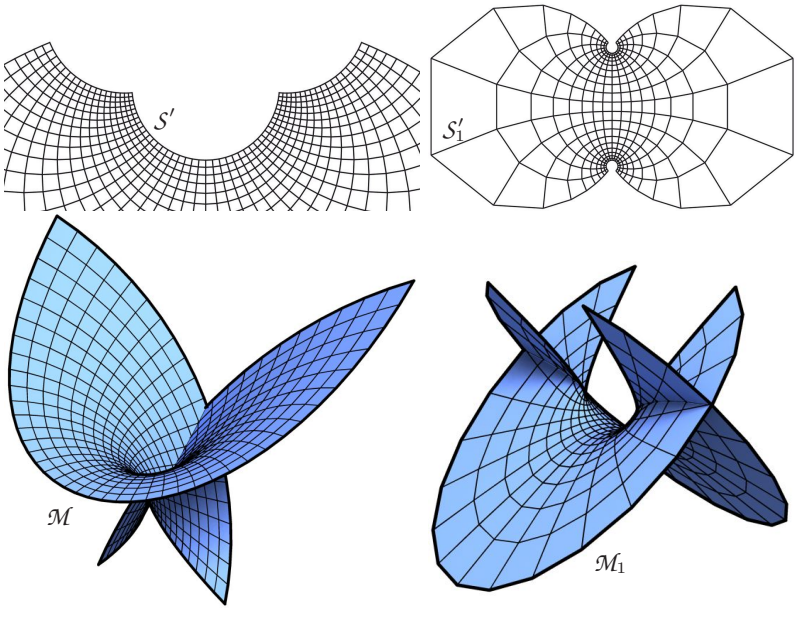


Fig. 8. Discrete i-minimal surfaces represented by i-isothermic meshes: isotropic Enneper surface $\mathcal{M}$ and isotropic counterpart $\mathcal{M}_1$ to Bonnet's minimal surface; the top views of the corresponding isotropic Koebe meshes are $\mathcal{S}', \mathcal{S}'_1$. In meshes $\mathcal{S}, \mathcal{M}, \mathcal{S}_1, \mathcal{M}_1$ all row and column polygons are planar.

a special i-isothermic mesh. This is the isotropic counterpart to the discrete Euclidean minimal surfaces of Bobenko et al. [2].

Figures 8 and 9 show some examples which have been generated as follows: Apply an inversion to the planar regular square grid (viewed as set of circles inscribed to the grid cells) to get a circle packing and a Koebe mesh $\mathcal{S}'$ in the plane. This is a discrete version of two orthogonal pencils of circles. Then lift

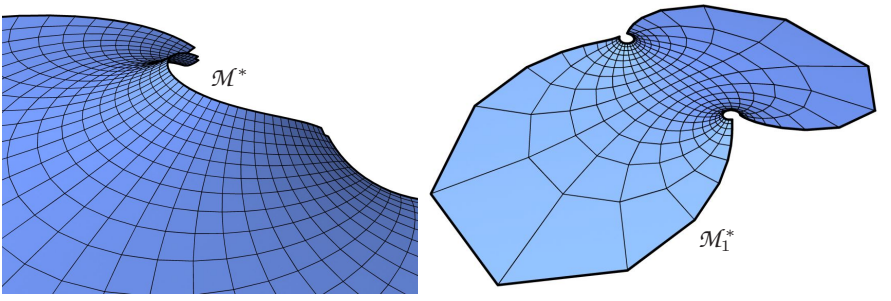


Fig. 9. Discrete i-minimal surfaces which are dual to the surfaces from Fig. 8

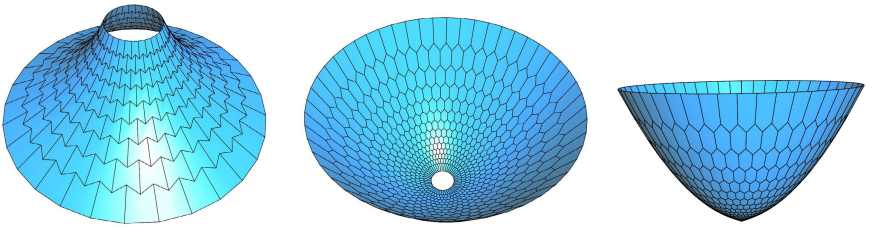


Fig. 10. Hexagonal meshes with planar faces and rotational symmetry, representing an i-minimal surface (left), a surface with constant i-mean curvature H (middle) and a surface with constant i-curvature K (right)

this mesh to the paraboloid Σ such that one obtains an i-Koebe mesh $\mathcal{S}$; the edges of $\mathcal{S}$ have to touch Σ. The Christoffel dual $\mathcal{M}$ of $\mathcal{S}$ is a discrete isotropic Enneper surface, whose smooth analogue has been studied by Strubecker [31]. Just lifting the square grid to Σ and then computing the Christoffel dual yields a discrete right hyperbolic paraboloid, the simplest i-minimal surface. Computing an i-Koebe mesh $\mathcal{S}_r$ with rotational symmetry and dualizing it, one obtains a discrete rotational i-minimal surface. Its smooth counterpart is the so-called logarithmoid, obtained by rotating the curve $z = \ln x$ of the xz-plane around the z-axis (for a hexagonal version, see Fig. 10, left). Finally, we may apply an isotropic Möbius transformation to $\mathcal{S}_r$ and obtain an i-Koebe mesh $\mathcal{S}_1$ whose top view $\mathcal{S}_1'$ discretizes two orthogonal pencils of circles through two points. The resulting i-minimal surface $\mathcal{M}_1$ is a discrete isotropic counterpart to Bonnet's minimal surface. The i-minimal meshes of this paragraph are exactly those which possess *planar discrete principal curvature lines* (row and column polygons).

Surfaces $\mathcal{M}^*$ and $\mathcal{M}_1^*$ which correspond to $\mathcal{M}$ and $\mathcal{M}_1$, resp., via metric duality are illustrated in Fig. 9. These surfaces have the property that the planes of all faces along the same row or column of the mesh pass through a fixed point (which may be at infinity).

The concept is not limited to quad meshes. We may also construct hexagonal meshes which are discrete versions of minimal surfaces, starting from hexagonal

Koebe polyhedra. In the example of Fig. 10, left, the realization of area$(F, S) = 0$ is simple due to rotational symmetry. It allows us to split each hexagon by the symmetry axis into two quads and compute the Christoffel dual. Fig. 10 also shows rotational hexagonal meshes whose discrete curvature measures H or K are constant; their construction is based on formula (17) and the exploitation of symmetry.

Remark 4. Application of an i-M-inversion κ to the inscribed circles of the faces in an i-minimal mesh $\mathcal{M}$ results in a collection of inscribed circles of the faces of another mesh $\kappa(\mathcal{M})$. The latter is no longer minimal, but can be shown to discretize the graph $z = f(x, y)$ of a biharmonic function ($\Delta^2 f = 0$), also known as an Airy surface [30]. By mapping $\mathcal{M}$ or $\kappa(\mathcal{M})$ back to the standard model of Laguerre geometry in the way explained in Remark 2, one obtains discrete versions of so-called Laguerre minimal surfaces (of the spherical type), which have been investigated by Blaschke [1]. Since the presented i-minimal meshes $\mathcal{M}$ are i-isothermic, the resulting Laguerre-minimal meshes $\mathcal{N}$ possess the edge offset property (cf. Remark 2). There are further types of Laguerre-minimal edge offset meshes, all of which can be constructed from Koebe meshes. These remarkable discrete surfaces will be the subject of a forthcoming publication.

I-curvatures at vertices. Metric duality in I^3 allows us to define discrete curvatures at vertices as well. Dual to the points of a domain F in a plane P is a set F^* of planes through a point P^*. By duality, the *isotropic measure (density) $D(F^*)$ of the plane set F^** in the sense of integral geometry [23] equals the isotropic area of F (area(F') of the top view F'). Let $P_0, \dots, P_{k-1}$ be the face planes of a mesh $\mathcal{M}$ around vertex $\mathbf{v}$. Each plane P_j can be written as $z = p_{j,1}x + p_{j,2}y + p_{j,3}$ and we set $\mathbf{p}_j := (p_{j,1}, p_{j,2})$. Then, the measure $D(\mathbf{v})$ of the 'vertex planes around $\mathbf{v}$' (which are dual to the points in the face $\mathbf{v}^*$ of the dual mesh $\mathcal{M}^*$) is computed as the areas in (15),

$$D(\mathbf{v}) = \frac{1}{2} \sum_{j=0}^{k-1} \det(\mathbf{p}_j, \mathbf{p}_{j+1}). \tag{21}$$

The rest follows easily: We consider a v-parallel mesh $\mathcal{S}$ of $\mathcal{M}$ which approximates Σ. Let $\mathbf{s} \in \mathcal{S}$ be the vertex corresponding to $\mathbf{v} \in \mathcal{M}$ (i.e., $\mathbf{s}' = \mathbf{v}'$). The planes through $\mathbf{s}$ possess a density measure $D(\mathbf{s}) = (1/2) \sum \det(\mathbf{q}_j, \mathbf{q}_{j+1})$ and analogous to (16) we define the *mixed measure* $D(\mathbf{v}, \mathbf{s}) = (1/4) \sum [\det(\mathbf{p}_j, \mathbf{q}_{j+1}) + \det(\mathbf{q}_j, \mathbf{p}_{j+1})]$. In view of equations (11) and (17) *curvatures of $\mathcal{M}$ at vertex $\mathbf{v}$ are defined by*

$$K(\mathbf{v}) = \frac{D(\mathbf{v})}{D(\mathbf{s})}, \quad H(\mathbf{v}) = \frac{D(\mathbf{v}, \mathbf{s})}{D(\mathbf{s})}. \tag{22}$$

Discrete face curvatures do not make much sense for triangle meshes $\mathcal{M}$ since f-parallel triangle meshes are similar and thus one cannot find an appropriate Gaussian image $\mathcal{S}$ unless $\mathcal{M}$ itself approximates Σ. However, vertex curvatures work for triangle meshes as well (see Fig. 11). In fact, here it is particularly easy to get $\mathcal{S}$: We project the vertices of $\mathcal{M}$ in isotropic z-direction onto Σ. The

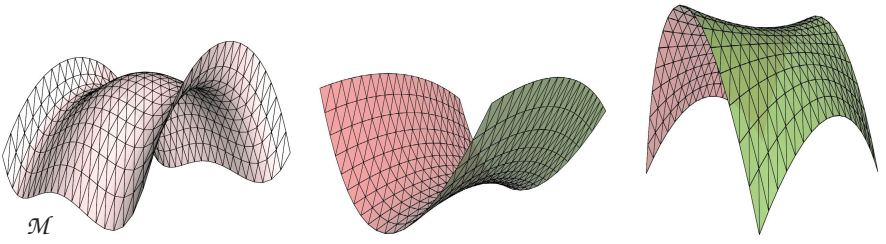


Fig. 11. Diagram surfaces for vertex curvatures H (middle) and K (right), computed with formula (22) for the triangle mesh $\mathcal{M}$, whose vertices lie on a given analytic surface $z = f(x, y)$. The curvatures match the true values with high accuracy.

relation between face curvatures, vertex curvatures and curvatures associated with edges of principal meshes will be described in a separate publication.

Conclusion and future research. The main contribution of the present paper is the geometric discussion of remarkable new types of meshes. For applications, i-minimal surfaces are certainly interesting. They deserve a much more detailed investigation and the derivation of tools such as the computation of a discrete i-minimal surface through a prescribed closed boundary.

Discrete differential geometry in its interplay with architecture is a wide area for future work. We only address a few topics related to isotropic geometry: Due to their optimal node properties, we need to know more about the possible shapes of quadrilateral edge offsets meshes and the related i-isothermic meshes. A careful investigation of smooth and discrete Laguerre-minimal surfaces and their suitability for architecture is also missing. Discrete surfaces in I^3 with constant H or K are also of interest.

Probably the most important research task towards the implementation of freeform geometry in architecture is to find new ways of approximating a given freeform shape by meshes with properties that are desirable for architecture (for a result obtained by currently available methods, see Fig. 5, bottom).

Acknowledgments. This research has been supported by grants No. S9206-N12 and No. P19214-N18 of the Austrian Science Fund (FWF). Special thanks go to Heinz Schmiedhofer who implemented the arrangement of supporting beams for architectural designs according to [21] and created most of the figures.

References

1. Blaschke, W.: Vorlesungen über Differentialgeometrie. vol. 3, Springer, Heidelberg (1929)
2. Bobenko, A., Hoffmann, T., Springborn, B.: Minimal surfaces from circle patterns: Geometry from combinatorics. Ann. of Math. 164, 231–264 (2006)
3. Bobenko, A., Pinkall, U.: Discrete isothermic surfaces. J. Reine Angew. Math. 475, 187–208 (1996)

4. Bobenko, A., Suris, Y.: Discrete differential geometry. Consistency as integrability. (2005), monograph pre-published at http://arxiv.org/abs/math.DG/0504358
5. Bobenko, A., Suris, Y.: On organizing principles of discrete differential geometry, geometry of spheres (to appear)
6. Bobenko, A., Springborn, B.: Variational principles for circle patterns and Koebe's theorem. Trans. Amer. Math. Soc. 356, 659–689 (2004)
7. Brell-Cokcan, S., Pottmann, H.: Tragstruktur für Freiformflächen in Bauwerken (Patent No. A1049/2006) (2006)
8. Cecil, T.: Lie Sphere Geometry. Springer, Heidelberg (1992)
9. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. ACM Trans. Graphics 23(3), 905–914 (2004)
10. Cutler, B., Whiting, E.: Constrained planar remeshing for architecture. In: Symp. Geom. Processing. poster (2006)
11. Glymph, J., et al.: A parametric strategy for freeform glass structures using quadrilateral planar facets. In: Glymph, J. (ed.) Acadia 2002, pp. 303–321. ACM Press, New York (2002)
12. Koenderink, I.J., van Doorn, A.J.: Image processing done right. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 158–172. Springer, Heidelberg (2002)
13. Liu, Y., Pottmann, H., Wallner, J., Yang, Y.L., Wang, W.: Geometric modeling with conical meshes and developable surfaces. ACM Trans. Graphics 25(3), 681–689 (2006)
14. Martin, R., de Pont, J., Sharrock, T.: Cyclide surfaces in computer aided design. In: Gregory, J.A. (ed.) The mathematics of surfaces, pp. 253–268. Clarendon Press, Oxford (1986)
15. Peternell, M.: Developable surface fitting to point clouds. Comp. Aid. Geom. Des. 21(8), 785–803 (2004)
16. Peternell, M., Pottmann, H.: A Laguerre geometric approach to rational offsets. Comp. Aid. Geom. Des. 15, 223–249 (1998)
17. Pottmann, H., Opitz, K.: Curvature analysis and visualization for functions defined on Euclidean spaces or surfaces. Comp. Aid. Geom. Des. 11, 655–674 (1994)
18. Pottmann, H., Peternell, M.: Applications of Laguerre geometry in CAGD. Comp. Aid. Geom. Des. 15, 165–186 (1998)
19. Pottmann, H., Brell-Cokcan, S., Wallner, J.: Discrete surfaces for architectural design. In: Curve and Surface Design: Avignon 2006, pp. 213–234. Nashboro Press (2006)
20. Pottmann, H., Wallner, J.: The focal geometry of circular and conical meshes. Adv. Comput. Math (to appear)
21. Pottmann, H., Liu, Y., Wallner, J., Bobenko, A., Wang, W.: Geometry of multilayer freeform structures for architecture. ACM Trans. Graphics 26(3) (2007)
22. Sachs, H.: Isotrope Geometrie des Raumes. Vieweg (1990)
23. Santalo, L.: Integral Geometry and Geometric Probability. Addison-Wesley, London (1976)
24. Sauer, R.: Differenzengeometrie. Springer, Heidelberg (1970)
25. Schneider, R.: Convex bodies: the Brunn- Minkowski theory. Cambridge University Press, Cambridge (1993)
26. Schramm, O.: Circle patterns with the combinatorics of the square grid. Duke Math. J. 86, 347–389 (1997)
27. Schramm, O.: How to cage an egg. Invent. Math. 107, 543–560 (1992)

28. Simon, U., Schwenck-Schellschmidt, A., Viesel, H.: Introduction to the affine differential geometry of hypersurfaces. Lecture Notes. Science Univ. Tokyo, Tokyo (1992)
29. Strubecker, K.: Differentialgeometrie des isotropen Raumes III: Flächentheorie. Math. Zeitschrift 48, 369–427 (1942)
30. Strubecker, K.: Airy'sche Spannungsfunktion und isotrope Differentialgeometrie. Math. Zeitschrift 78, 189–198 (1962)
31. Strubecker, K.: Über das isotrope Gegenstück $z = \frac{3}{2}J(x+iy)^{2/3}$ der Minimalfläche von Enneper. Abh. Math. Sem. Univ. Hamburg 44, 152–174 (1975/76)
32. Strubecker, K.: Duale Minimalflächen des isotropen Raumes. Rad JAZU 382, 91–107 (1978)
33. Wallner, J., Pottmann, H.: Infinitesimally flexible meshes and discrete minimal surfaces. Monatsh. Math. (to appear)
34. Ziegler, G.: Lectures on Polytopes. Springer, Heidelberg (1995)

An Appropriate Geometric Invariant for the C^2-Analysis of Subdivision Surfaces

Ulrich Reif

Dept. of Mathematics, Darmstadt Univeristy of Technology, Germany
reif@mathematik.tu-darmstadt.de

Abstract. We introduce the embedded Weingarten map as a geometric invariant of piecewise smooth surfaces. It is given by a (3×3)-matrix and provides complete curvature information in a continuous way. Thus, it is the appropriate tool for the C^2-analysis of subdivision surfaces near extraordinary points. We derive asymptotic expansions and show that the convergence of the sequence of embedded Weingarten maps to a constant limit is necessary and sufficient for curvature continuity.

1 Introduction

Locally, a subdivision surface $\mathbf{x}$ can be regarded as the union of the *central point* $\mathbf{x}^{\mathrm{c}}$ and a sequence of smaller and smaller *spline rings* $\mathbf{x}^m, m \in \mathbb{N}$, which are converging to the *central point* $\mathbf{x}^{\mathrm{c}}$,

$$\mathbf{x}^{\mathrm{c}} = \lim_{m \to \infty} \mathbf{x}^m.$$

While, at least for linear stationary schemes, the parametrizations of the $\mathbf{x}^m$ are explicitly known, the central point $\mathbf{x}^{\mathrm{c}}$ itself is an isolated object so that standard formulas for computing normal vector and curvature properties at this point cannot be applied immediately. There are two ways to analyze $\mathbf{x}$ in a vicinity of $\mathbf{x}^{\mathrm{c}}$:

First, one can try to express $\mathbf{x}$ locally as the graph of a real-valued height function h over the tangent plane at $\mathbf{x}^{\mathrm{c}}$. Then h contains all curvature information of $\mathbf{x}$ at $\mathbf{x}^{\mathrm{c}}$. In particular, by definition, $\mathbf{x}$ is *curvature continuous* at $\mathbf{x}^{\mathrm{c}}$ if and only if h is C^2. Beyond its fundamental character as a definition, this approach can also be used for analytical purposes. For instance, in [1], convergence of the *anchored osculating paraboloid*, which is just the quadratic Taylor jet of h, is used to prove curvature continuity of surfaces generated by guided subdivision. However, explicit formulas for h and its derivatives are not easy to derive since a nonlinear pair of equations has to be solved in order to determine the value of h at a given point in the tangent plane.

Second, one can analyze convergence properties of geometric invariants to draw conclusions on $\mathbf{x}$ at $\mathbf{x}^{\mathrm{c}}$. For instance, $\mathbf{x}$ is normal continuous at $\mathbf{x}^{\mathrm{c}}$ if the sequence of normal vectors $\mathbf{n}^m$ of the spline rings is converging to a unique limit, called the *central normal*,

$$\mathbf{n}^{\mathrm{c}} := \lim_{m \to \infty} \mathbf{n}^m.$$

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 364–377, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Equally, it is suggested in [4] to establish curvature continuity based on the convergence of the principal curvatures $\kappa_{1,2}^m$ and the principal directions $\mathbf{r}_{1,2}^m$, which are the standard second order geometric invariants of smooth surfaces,

$$\kappa_{1,2}^{\mathrm{c}} := \lim_{m\to\infty} \kappa_{1,2}^m, \quad \mathbf{r}_{1,2}^{\mathrm{c}} := \lim_{m\to\infty} \mathbf{r}_{1,2}^m.$$

Favorably, these quantities are easy to derive from the underlying parametrizations of the spline rings $\mathbf{x}^m$. However, there is a nasty little problem with this approach: if $\mathbf{x}^{\mathrm{c}}$ happens to be an *umbilic point* of the surface, i.e., $\kappa_1^{\mathrm{c}} = \kappa_2^{\mathrm{c}}$, then the corresponding principal directions are *not* convergent at $\mathbf{x}^{\mathrm{c}}$, even if the surface is curvature continuous. The reason for that phenomenon is the fact that at umbilic points any direction in the tangent plane is a principal direction. Consequently, we have to state that curvature analysis based on principal directions is a flawed approach which needs mending.

The principal directions are the eigenvectors of the *Weingarten map*, also called the *shape operator*, to the principal curvatures. Hence, one could try to consider convergence properties of the sequence W^m of Weingarten maps instead of principal curvatures and directions. In principle, this is possible, but point-wise, the Weingarten map lives in the tangent space of the surface, and a coordinate system in this tangent space is required to express W^m explicitly. A standard choice for such a coordinate system is given by the partial derivatives of the surface parametrization. This parametrization, however, is typically not C^1 for a subdivision surface, even if it is geometrically smooth. Hence, with respect to these coordinates, the Weingarten map does not vary continuously on the surface. A continuous choice of coordinates might locally be difficult, and, for topological reasons, globally be impossible.

In this paper, we propose to extend the Weingarten map to the embedding space $\mathbb{R}^3$. Besides the principal directions, now regarded as vectors in $\mathbb{R}^3$, also the surface normal becomes an eigenvector corresponding to the trivial eigenvalue 0. The advantages of the *embedded Weingarten map*, which make it appropriate for the study of subdivision surfaces, are that

- it is a *geometric invariant* of the surface, i.e., it does not depend on the chosen parametrization but only on the shape of the surface;
- it *varies continuously* on the surface (even at umbilic points) if and only if the surface is C^2;
- it is *easy to compute* using the second fundamental form and the pseudo-inverse of the derivative of the parametrization.

In Section 2, we develop the concept of the embedded Weingarten map in some detail. Then, in Section 3, we apply it to subdivision surfaces and derive asymptotic expansions for the sequence $\mathbf{W}^m$ of embedded Weingarten maps of the spline rings. Finally, in Section 4, the convergence requirements on the sequence $\mathbf{W}^m$ are used to recall the well known conditions on curvature continuity.

2 The Extended Weingarten Map

Throughout the paper, points and vectors in $\mathbb{R}^3$ are denoted by boldface latin letters and understood as *rows* with three components, e.g., $\mathbf{x} = [x, y, z]$. Points in $\mathbb{R}^2$, such as parameters of surfaces, are denoted by boldface greek letters, e.g., $\boldsymbol{\sigma} = (s, t)$. For matrices A, B, a dot indicates multiplication by the transpose,

$$A \cdot B := AB^{\mathrm{t}}.$$

Equally, if one or both factors are points in $\mathbb{R}^3$, we define $A \cdot \mathbf{x} := A\mathbf{x}^{\mathrm{t}}$, and $\mathbf{x} \cdot \mathbf{y} := \mathbf{x}\mathbf{y}^{\mathrm{t}}$. The Euclidean norm is denoted $|\mathbf{x}| := \sqrt{\mathbf{x} \cdot \mathbf{x}}$. We consider an at least twice differentiable parametrized surface

$$\mathbf{x} : \Omega \ni (s, t) \to \mathbf{x}(s, t) \in \mathbb{R}^3$$

with domain $\Omega \subset \mathbb{R}^2$. According to the above convention, we also write $\mathbf{x}(\boldsymbol{\sigma})$ instead of $\mathbf{x}(s, t)$, but mostly, the argument is simply omitted. The trace of $\mathbf{x}$ is denoted $\mathbf{X} := \mathbf{x}(\Omega)$. The first derivative of $\mathbf{x}$ is given by the (2×3)-matrix

$$D\mathbf{x} := \begin{bmatrix} \mathbf{x}_s \\ \mathbf{x}_t \end{bmatrix},$$

and we assume that the cross product of its rows does not vanish,

$${}^{\times}D\mathbf{x} := \mathbf{x}_s \times \mathbf{x}_t \neq \mathbf{0}.$$

In other words, $\mathbf{x}$ is a *regular C^2-surface*. The *Gauss map* $\mathbf{n}$ of $\mathbf{x}$ assigns to each parameter $\boldsymbol{\sigma}$ the corresponding normalized *normal vector*,

$$\mathbf{n} : \Omega \ni \boldsymbol{\sigma} \to \frac{{}^{\times}D\mathbf{x}(\boldsymbol{\sigma})}{|{}^{\times}D\mathbf{x}(\boldsymbol{\sigma})|} \in S^2.$$

Differentiating the identity $\mathbf{n} \cdot \mathbf{n} \equiv 1$, we obtain $D\mathbf{n} \cdot \mathbf{n} = 0$, what means that the row vectors of $D\mathbf{n}$ lie in the tangent plane of $\mathbf{x}$ at the corresponding point. Hence, by regularity, there exists a (2×2)-matrix W, called the *Weingarten map*, with

$$D\mathbf{n} = -W D\mathbf{x}.$$

Multiplication with the transpose of $D\mathbf{x}$ yields

$$I\!I = W I,$$

where

$$I := D\mathbf{x} \cdot D\mathbf{x} = \begin{bmatrix} \mathbf{x}_s \cdot \mathbf{x}_s & \mathbf{x}_t \cdot \mathbf{x}_s \\ \mathbf{x}_s \cdot \mathbf{x}_t & \mathbf{x}_t \cdot \mathbf{x}_t \end{bmatrix}$$

and also

$$I\!I := -D\mathbf{n} \cdot D\mathbf{x} = -\begin{bmatrix} \mathbf{x}_s \cdot \mathbf{n}_s & \mathbf{x}_t \cdot \mathbf{n}_s \\ \mathbf{x}_s \cdot \mathbf{n}_t & \mathbf{x}_t \cdot \mathbf{n}_t \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{ss} \cdot \mathbf{n} & \mathbf{x}_{st} \cdot \mathbf{n} \\ \mathbf{x}_{st} \cdot \mathbf{n} & \mathbf{x}_{tt} \cdot \mathbf{n} \end{bmatrix}$$

are symmetric (2×2)-matrices, called the *first* and *second fundamental form of* $\mathbf{x}$, respectively. By $\det I = |{}^{\times}D\mathbf{x}|^2$ and regularity of the parametrization, I is invertible, thus

$$W = II\, I^{-1}.$$

Given a smooth curve $\boldsymbol{\gamma}(t) = \boldsymbol{\sigma} + t\boldsymbol{\sigma}' + o(t)$ in the domain of $\mathbf{x}$, we define the related curves

$$\mathbf{c}_{\mathbf{x}}(t) := \mathbf{x}(\boldsymbol{\gamma}(t)), \quad \mathbf{c}_{\mathbf{n}}(t) := \mathbf{n}(\boldsymbol{\gamma}(t))$$

on the surface and on the unit sphere, respectively. Dropping, as usual, the parameter $\boldsymbol{\sigma} = \boldsymbol{\gamma}(0)$, we obtain by the chain rule

$$\mathbf{c}'_{\mathbf{x}}(0) = \boldsymbol{\sigma}' D\mathbf{x}, \quad \mathbf{c}'_{\mathbf{n}}(0) = -\boldsymbol{\sigma}' W D\mathbf{x}.$$

Now, we are looking for curves $\boldsymbol{\gamma}$ with the property that

- $\mathbf{r} := \mathbf{c}'_{\mathbf{x}}(0)$ has unit length, and
- $\mathbf{c}'_{\mathbf{n}}(0) = -\kappa\mathbf{r}$ for some $\kappa \in \mathbb{R}$, i.e., $\mathbf{c}'_{\mathbf{x}}(0)$ and $\mathbf{c}'_{\mathbf{n}}(0)$ are parallel.

Then the vector $\mathbf{r} \in \mathbb{R}^3$ is called a *principal direction*, and κ is the corresponding *principal curvature* of $\mathbf{x}$ at the point $\mathbf{x}(\boldsymbol{\sigma})$. The condition $\mathbf{c}'_{\mathbf{n}}(0) = \kappa\mathbf{r}$ is equivalent to

$$\boldsymbol{\sigma}' W = \kappa\boldsymbol{\sigma}', \quad \mathbf{r} = \boldsymbol{\sigma}' D\mathbf{x}.$$

In other words, $\boldsymbol{\sigma}'$ is a left eigenvector of W with eigenvalue κ. One can (and we will) show that W has always two real eigenvalues κ_1, κ_2, and that the corresponding pair $\mathbf{r}_1, \mathbf{r}_2$ of principal directions can be chosen orthonormal. It is well known that the principal curvatures and directions are *geometric invariants* in the sense that they do not depend on the parametrization, but only on the shape and the orientation of the surface. We state without proof

Theorem 1. *Let* $\mathbf{x}$ *and* $\tilde{\mathbf{x}}$ *be two regular* C^2*-surfaces without self-intersections and equal trace,* $\mathbf{X} = \tilde{\mathbf{X}}$*. Then, for any* $\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}}$ *with* $\mathbf{x}(\boldsymbol{\sigma}) = \tilde{\mathbf{x}}(\tilde{\boldsymbol{\sigma}})$*,*

$$\mathbf{n}(\boldsymbol{\sigma}) = s\tilde{\mathbf{n}}(\tilde{\boldsymbol{\sigma}})$$

either for $s = 1$ *or* $s = -1$*. Further, if* $\mathbf{r}$ *is a principal direction of* $\mathbf{x}$ *at* $\mathbf{x}(\boldsymbol{\sigma})$ *with the principal curvature* κ*, then it also a principal direction of* $\tilde{\mathbf{x}}$ *at* $\tilde{\mathbf{x}}(\tilde{\boldsymbol{\sigma}})$ *with the principal curvature* $s\kappa$*.*

The theory developed so far is well established, but not fully satisfactory for the analysis of subdivision surfaces. Clearly, if we assume for instance $\kappa_1 \leq \kappa_2$, the principal curvatures depend continuously on the parameter $\boldsymbol{\sigma} \in \Omega$. But by contrast, the principal directions reveal discontinuities at *umbilic points* which are characterized by $\kappa_1 = \kappa_2$. Here, any direction in the tangent plane is a principal direction, and accordingly, $\mathbf{r}_1, \mathbf{r}_2$ do not converge when approaching such a point. Below, this phenomenon is illustrated at hand of a paraboloid of revolution, where the principal directions diverge near the vertex. In standard text books on differential geometry, this problem is not fully resolved, but merely addressed as a degenerate situation. In our context, however, continuity is an

indispensable analytic tool since we want to study convergence properties of spline rings without having immediate access to a single surface parametrization near the extraordinary point. We suggest the following approach:

Definition 2. *For a regular C^2-surface* $\mathbf{x}$*, let*

$$D\mathbf{x}^+ := (I^{-1} D\mathbf{x})^{\mathrm{t}}$$

denote the pseudo-inverse of $D\mathbf{x}$*, which is a* (3×2)*-matrix. Then we define the* embedded Weingarten map *of* $\mathbf{x}$ *as the symmetric* (3×3)*-matrix*

$$\mathbf{W} := D\mathbf{x}^+ \, II \cdot D\mathbf{x}^+.$$

We claim that this object is a geometric invariant which contains the complete curvature information in a continuous way. Because $\mathbf{W} \cdot \mathbf{n} = \mathbf{0}$, the normal vector is always an eigenvector of $\mathbf{W}$ with eigenvalue 0. The two other eigenvectors can be chosen orthonormal (mutually and with respect to $\mathbf{n}$), and collected in a (2×3)-matrix $\mathbf{R}$. The diagonal matrix of the corresponding pair of eigenvalues is denoted K. We obtain the factorization

$$\mathbf{W} = [\mathbf{R}^{\mathrm{t}} \, \mathbf{n}^{\mathrm{t}}] \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{n} \end{bmatrix} = \mathbf{R}^{\mathrm{t}} K \mathbf{R},$$

and therefore

$$\mathbf{R}\mathbf{W} = K\mathbf{R}.$$

By orthogonality of the eigenvectors, we have $\mathbf{R} \cdot \mathbf{n} = \mathbf{0}$. Hence, there exists a (2×2)-matrix Σ with $\mathbf{R} = \Sigma D\mathbf{x}$. Together with the definitions $\mathbf{W} = D\mathbf{x}^{\mathrm{t}} I^{-1} II I^{-1} D\mathbf{x}$ and $W = II \, I^{-1}$, we conclude from the last display $\Sigma W D\mathbf{x} = W \Sigma D\mathbf{x}$, and eventually

$$\Sigma W = K \Sigma.$$

That is, the diagonal entries of $K = \mathrm{diag}(\kappa_1, \kappa_2)$ are the eigenvalues of W, and the rows of $\Sigma = [\boldsymbol{\sigma}_1'; \boldsymbol{\sigma}_2']$ are the corresponding left eigenvectors which, via $[\mathbf{r}_1; \mathbf{r}_2] := \Sigma D\mathbf{x} = \mathbf{R}$, yield the principal directions.

So far, we have found the following: The embedded Weingarten map $\mathbf{W}$ is a symmetric (3×3)-matrix with a trivial eigenvalue 0 corresponding to the surface normal $\mathbf{n}$. The other two eigenvalues are the principal curvatures, and the corresponding eigenvectors are the principal directions of the surface. But unlike these directions, the matrix $\mathbf{W}$ depends continuously on the parameter $\boldsymbol{\sigma}$. The following theorem establishes $\mathbf{W}$ as a geometric invariant:

Theorem 3. *Let* $\mathbf{x}$ *and* $\tilde{\mathbf{x}}$ *be two regular C^2-surfaces without self-intersections and equal trace,* $\mathbf{X} = \tilde{\mathbf{X}}$*, and let* $\mathbf{n}(\boldsymbol{\sigma}) = s\tilde{\mathbf{n}}(\tilde{\boldsymbol{\sigma}})$ *for* $\mathbf{x}(\boldsymbol{\sigma}) = \tilde{\mathbf{x}}(\tilde{\boldsymbol{\sigma}})$ *as in Theorem 1. Then the corresponding embedded Weingarten maps are equal up to sign,*

$$\mathbf{W}(\boldsymbol{\sigma}) = s\tilde{\mathbf{W}}(\tilde{\boldsymbol{\sigma}}).$$

Proof. Let $s = 1$. If $\mathbf{r}$ is a principal direction of $\mathbf{x}$ at $\mathbf{x}(\boldsymbol{\sigma})$ to the principal curvature κ, then, by Theorem 1, it is also a principal direction of $\tilde{\mathbf{x}}$ at $\tilde{\mathbf{x}}(\tilde{\boldsymbol{\sigma}})$ to κ. Further, $\mathbf{W}(\boldsymbol{\sigma}) \cdot \mathbf{n}(\boldsymbol{\sigma}) = \tilde{\mathbf{W}}(\tilde{\boldsymbol{\sigma}}) \cdot \tilde{\mathbf{n}}(\tilde{\boldsymbol{\sigma}}) = \mathbf{0}$. Hence, the eigenspaces and eigenvalues of $\mathbf{W}(\boldsymbol{\sigma})$ and $\tilde{\mathbf{W}}(\tilde{\boldsymbol{\sigma}})$ coincide so that $\mathbf{W}(\boldsymbol{\sigma}) = \tilde{\mathbf{W}}(\tilde{\boldsymbol{\sigma}})$. If $s = -1$, then again, the corresponding eigenspaces coincide. However, the eigenvalues and hence the matrices have opposite sign. □

The mean and the product of the principal curvatures are known as *mean curvature* and *Gaussian curvature* of $\mathbf{x}$, respectively,

$$K_m := \frac{\kappa_1 + \kappa_2}{2}, \quad K_g := \kappa_1 \kappa_2.$$

They can be computed easily from W,

$$K_m = \frac{1}{2} \operatorname{trace} W, \quad K_g = \det W = \det II / \det I,$$

and equally from $\mathbf{W}$,

$$K_m = \frac{1}{2} \operatorname{trace} \mathbf{W}, \quad K_g = \frac{1}{2} \operatorname{trace}^2 \mathbf{W} - \frac{1}{2} \|\mathbf{W}\|_{\mathrm{F}}^2,$$

where $\|\mathbf{W}\|_{\mathrm{F}}^2 := \sum_{i,j} \mathbf{W}_{i,j}^2$ is the squared Frobenius norm of $\mathbf{W}$. A point $\mathbf{x}(\boldsymbol{\sigma})$ is called *elliptic* if $K_g(\boldsymbol{\sigma}) > 0$, *hyperbolic* if $K_g(\boldsymbol{\sigma}) < 0$, and *parabolic* if $K_g(\boldsymbol{\sigma}) = 0$. Now, we consider a continuous surface $\mathbf{x} : \Omega \to \mathbb{R}^3$ which is regular and C^k everywhere with the exception of a single point, say $\boldsymbol{\sigma}_0 = \mathbf{0}$. Such an $\mathbf{x}$ is called a C_0^k-surface. More generally, we say that $\mathbf{x}$ is a C_r^k surface if there exists a regular C^r-surface $\tilde{\mathbf{x}}$ with equal trace, $\mathbf{X} = \tilde{\mathbf{X}}$. It is well known that $\mathbf{x}$ is a C_1^k-surface if the Gauss map converges,

$$\mathbf{n}^{\mathrm{c}} := \lim_{\boldsymbol{\sigma} \to \mathbf{0}} \mathbf{n}(\boldsymbol{\sigma}),$$

and if the projection of $\mathbf{x}$ to the tangent plane at $\mathbf{x}(\mathbf{0})$ is locally injective. The next theorem provides a similar result for curvature continuity and will be fundamental for the investigation of subdivision surfaces:

Theorem 4. *Let $\mathbf{x} : \Omega \to \mathbb{R}^3$ be a C_1^k-surface, $k \geq 2$. It is a C_2^k-surface if and only if the embedded Weingarten map converges, i.e., if the limit*

$$\mathbf{W}^{\mathrm{c}} = \lim_{\boldsymbol{\sigma} \to \mathbf{0}} \mathbf{W}(\boldsymbol{\sigma})$$

exists.

Proof. Without loss of generality, let us assume that $\mathbf{n}^{\mathrm{c}} = \mathbf{e}_3 := [0, 0, 1]$ is the third unit vector, and that $\mathbf{x}(\mathbf{0}) = \mathbf{0}$. Since the projection of $\mathbf{x}$ to the tangent plane at $\mathbf{x}(\mathbf{0})$ is locally injective, it can be represented as graph over the tangent plane. That is, there exists a scalar function h such that, locally,

$$\tilde{\mathbf{x}}(u, v) := [u, v, h(u, v)] = \mathbf{x}(\boldsymbol{\sigma}).$$

The function h is C^1, and, by the inverse function theorem, it is twice differentiable for $(u,v) \neq (0,0)$. By Theorem 3, the embedded Weingarten maps of $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are equal up to sign so that

$$\lim_{(u,v)\to(0,0)} \tilde{\mathbf{W}}(u,v) = s \lim_{\boldsymbol{\sigma}\to\boldsymbol{\sigma}_0} \mathbf{W}(\boldsymbol{\sigma}) = s\mathbf{W}^{\mathrm{c}}, \quad s \in \{-1,1\}.$$

Now, the mean value theorem shows that h is twice differentiable also at the origin, and that $s\mathbf{W}^{\mathrm{c}}$ is the embedded Weingarten map at this point. □

Let us illustrate the concepts developed so far at hand of a simple example. We consider the C_0^∞-surface

$$\mathbf{x}(s,t) := [2\sqrt{s},\ 2\sqrt{t},\ s+t], \quad (s,t) \in [0,1]^2.$$

The parametrization is not differentiable at the origin so that we do not know beforehand if the surface is normal or curvature continuous. Of course, the reparametrization $[x, y, (x^2+y^2)/4]$ shows that it is part of a paraboloid of revolution, which settles the problem immediately. But in more complicated situations, such knowledge is not readily available. With $w := (1+s+t)^{-1/2}$, we obtain for $(s,t) \neq (0,0)$

$$D\mathbf{x} = \begin{bmatrix} 1/\sqrt{s} & 0 & 1 \\ 0 & 1/\sqrt{t} & 1 \end{bmatrix}, \quad \mathbf{n} = w\,[-\sqrt{s},\ -\sqrt{t},\ 1].$$

The normal vector converges according to

$$\lim_{(s,t)\to(0,0)} \mathbf{n}(s,t) = [0,\ 0,\ 1],$$

and obviously, the projection of $\mathbf{x}$ to the xy-plane is injective so that $\mathbf{x}$ is a C_1^∞-surface. Further,

$$I = \begin{bmatrix} 1+1/s & 1 \\ 1 & 1+1/t \end{bmatrix}, \quad II = \frac{w}{2}\begin{bmatrix} 1/s & 0 \\ 0 & 1/t \end{bmatrix}, \quad W = \frac{w^3}{2}\begin{bmatrix} 1+t & -t \\ -s & 1+s \end{bmatrix}.$$

Hence, the principal curvatures are $\kappa_1 = w/2$ and $\kappa_2 = w^3/2$. For $(s,t) \neq (0,0)$, the corresponding left eigenvectors are unique up to orientation, and we obtain

$$\mathbf{r}_1 = \frac{1}{\sqrt{s+t}}\,[\sqrt{t},\ -\sqrt{s},\ 0], \quad \mathbf{r}_2 = \frac{w}{\sqrt{s+t}}\,[\sqrt{s},\ \sqrt{t},\ s+t].$$

Obviously, these vectors do *not* converge as $(s,t) \to (0,0)$. Rather, the origin is an umbilic point, and any direction in the xy-plane is a principal direction. Now we compute the embedded Weingarten map. With $v := 2+s+t$, we find for $(s,t) \neq (0,0)$

$$D\mathbf{x}^+ = w^2 \begin{bmatrix} \sqrt{s}(1+t) & -\sqrt{st} \\ -\sqrt{t}s & \sqrt{t}(1+s) \\ s & t \end{bmatrix}, \quad \mathbf{W} = \frac{w^5}{2}\begin{bmatrix} 1+vt & -v\sqrt{st} & \sqrt{s} \\ -v\sqrt{st} & 1+vs & \sqrt{t} \\ \sqrt{s} & \sqrt{t} & s+t \end{bmatrix}.$$

Its eigenvalues are $\kappa_1, \kappa_2, 0$, and the corresponding eigenvectors are $\mathbf{r}_1, \mathbf{r}_2, \mathbf{n}$. At the origin, we obtain the limit

$$\lim_{(s,t)\to(0,0)} \mathbf{W}(s,t) = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Unlike its first two eigenvectors, this matrix is uniquely determined. Because $\mathbf{W}$ depends continuously on (s,t) on the whole domain, we can conclude that $\mathbf{x}$ is a C_2^∞-surface, or in other words, it is curvature continuous.

One could argue that also the standard Weingarten map W converges as $(s,t) \to (0,0)$, and that curvature continuity follows equally from that. This is correct, but the situation changes if we attach to $\mathbf{x}$ a second piece of surface

$$\tilde{\mathbf{x}}(s,t) := [-2\sqrt{t},\ 2\sqrt{s},\ s+t], \quad (s,t) \in [0,1]^2$$

with

$$\tilde{\mathbf{n}} = w\,[-\sqrt{t},\ \sqrt{s},\ s+t],\ \tilde{W} = \frac{w^3}{2} \begin{bmatrix} 1+t & -t \\ -s & 1+s \end{bmatrix},\ \tilde{\mathbf{W}} = \frac{w^5}{2} \begin{bmatrix} 1+vs & v\sqrt{st} & -\sqrt{t} \\ v\sqrt{st} & 1+vt & \sqrt{s} \\ -\sqrt{t} & \sqrt{s} & s+t \end{bmatrix}.$$

Then $\mathbf{x}$ and $\tilde{\mathbf{x}}$ join normal continuous according to

$$\begin{aligned} \mathbf{x}(0,u) &= \tilde{\mathbf{x}}(u,0) = [0,\ 2\sqrt{u},\ 0], \quad u \in [0,1] \\ \mathbf{n}(0,u) &= \tilde{\mathbf{n}}(u,0) = (1+u)^{-1/2}\,[0,\ -\sqrt{u},\ 1]. \end{aligned}$$

The corresponding standard Weingarten maps differ at the common boundary,

$$W(0,u) = \frac{(1+u)^{-3/2}}{2} \begin{bmatrix} 1+u & -u \\ 0 & 1 \end{bmatrix}, \quad \tilde{W}(u,0) = \frac{(1+u)^{-3/2}}{2} \begin{bmatrix} 1 & 0 \\ -u & 1+u \end{bmatrix},$$

while the embedded Weingarten maps coincide,

$$\mathbf{W}(0,u) = \tilde{\mathbf{W}}(u,0) = \frac{(1+u)^{-5/2}}{2} \begin{bmatrix} (1+u)^2 & 0 & 0 \\ 0 & 1 & \sqrt{u} \\ 0 & \sqrt{u} & u \end{bmatrix}.$$

From the latter observation, we conclude that the composed surface $\mathbf{X} \cup \tilde{\mathbf{X}}$ is curvature continuous.

The following sketchy considerations concerning the embedded Weingarten map are not directly related to the forthcoming analysis, but perhaps interesting in their own right:

– For a non-parabolic point, i.e, $K_g(\boldsymbol{\sigma}) \neq 0$, we define the pseudo-inverse of $\mathbf{W}$ by

$$\mathbf{W}^+ := \mathbf{R}^{\mathrm{t}} K^{-1} \mathbf{R}.$$

Then the embedded Weingarten map is related to the first and second fundamental form by

$$I = D\mathbf{n}\,(\mathbf{W}^+)^2 \cdot D\mathbf{n}, \quad II = D\mathbf{n}\,\mathbf{W}^+ \cdot D\mathbf{n}.$$

Further, $D\mathbf{n}$ and $D\mathbf{x}$ are related by

$$D\mathbf{n} = -D\mathbf{x}\,\mathbf{W}, \quad D\mathbf{x} = -D\mathbf{n}\,\mathbf{W}^+.$$

Differentiating the last equation with respect to s and t, we find that the *integrability condition* $\mathbf{x}_{st} = \mathbf{x}_{ts}$ is equivalent to

$$\mathbf{n}_s \mathbf{W}_t^+ = \mathbf{n}_t \mathbf{W}_s^+,$$

provided that $\mathbf{n}_{st} = \mathbf{n}_{ts}$. Compared with that, the integrability conditions for the first and second fundamental form, also known as the Mainardi-Codazzi and Gauss equations, are slightly more complicated.

– If, as before, near a point $\mathbf{x}(\boldsymbol{\sigma})$ the surface $\mathbf{x}$ is locally described as the graph of a local height function h, and if coordinates are chosen such that $\mathbf{x}(\boldsymbol{\sigma}) = \mathbf{0}$ is the origin and the surface normal is the third unit vector, then the quadratic Taylor jet

$$h_2(u, v) = au^2 + 2buv + cv^2 \tag{1}$$

of h is called the *osculating paraboloid* of $\mathbf{x}$ at $\mathbf{x}(\boldsymbol{\sigma})$. This surface can be regarded as a quadric, which is degenerate in the sense that at least one eigenvalue vanishes. Its vertex is the origin, the two main axes in the xy-plane are the principal directions of $\mathbf{x}$, and the corresponding eigenvalues are the principal curvatures. Further, its symmetry axis, which is the main axes corresponding to the eigenvalue 0, is the normal vector $\mathbf{n}(\boldsymbol{\sigma})$. These informations are sufficient to verify that, in implicit form, the osculating paraboloid is the set of all $\mathbf{y} \in \mathbb{R}^3$ satisfying

$$\big((\mathbf{y} - \mathbf{x})\mathbf{W} - 2\mathbf{n}\big) \cdot (\mathbf{y} - \mathbf{x}) = 0, \tag{2}$$

where of course, $\mathbf{x}, \mathbf{n}$ and $\mathbf{W}$ are evaluated at $\boldsymbol{\sigma}$. The advantage of (2) over (1) is that it does not refer to special coordinates.

– Let $\mathbf{X}$ be a surface given in implicit form,

$$\mathbf{X} := \{\mathbf{y} \in \mathbb{R}^3 : f(\mathbf{y}) = 0\}.$$

A regular point $\mathbf{x} \in \mathbf{X}$ of the surface is characterized by

$$f(\mathbf{x}) = 0, \quad \mathbf{g} := \nabla f(\mathbf{x}) \neq \mathbf{0}.$$

Then $\mathbf{n} := -\mathbf{g}/|\mathbf{g}|$ is the normal vector (or more precisely one of the two normal vectors) of $\mathbf{X}$ at $\mathbf{x}$. The quadratic Taylor expansion of $2f$ at $\mathbf{x}$ reads

$$f_2(\mathbf{y}) = \big((\mathbf{y} - \mathbf{x})H + 2\mathbf{g}\big) \cdot (\mathbf{y} - \mathbf{x}),$$

where H is the Hessian H of f at $\mathbf{x}$. Hence, the quadric $\mathbf{Q} := \{\mathbf{y} : f_2(\mathbf{y}) = 0\}$ is osculating $\mathbf{X}$ at $\mathbf{x}$. Now let us consider the modified function

$$\tilde{f}_2(\mathbf{y}) = \big((\mathbf{y}-\mathbf{x})\tilde{H} + 2\mathbf{g}\big) \cdot (\mathbf{y}-\mathbf{x}),$$

where

$$\tilde{H} := (\mathrm{Id} - N)H(\mathrm{Id} - N), \quad N := \mathbf{n}^{\mathrm{t}}\mathbf{n},$$

yielding the quadric $\tilde{\mathbf{Q}} := \{\mathbf{y} : \tilde{f}_2(\mathbf{y}) = 0\}$. First, we observe that also $\tilde{\mathbf{Q}}$ is osculating $\mathbf{X}$ at $\mathbf{x}$ because $(H - \tilde{H}) \cdot \mathbf{t} = 0$ for all vectors $\mathbf{t}$ with $\mathbf{t} \cdot \mathbf{n} = 0$. Second, $\tilde{\mathbf{Q}}$ is a paraboloid with symmetry axis $\mathbf{n}$ because $\tilde{H} \cdot \mathbf{n} = 0$. Hence, $\tilde{\mathbf{Q}}$ is the osculating paraboloid of $\mathbf{X}$ at $\mathbf{x}$, and comparison with (2), using $\mathbf{g} = -|\mathbf{g}|\mathbf{n}$, shows that

$$\mathbf{W} := \frac{\tilde{H}}{|\mathbf{g}|}$$

is the embedded Weingarten map of $\mathbf{X}$ at $\mathbf{x}$. The formulas derived here can also be found in the unpublished manuscript [5].

3 Asymptotic Expansions

In this section, we derive asymptotic expansions for the fundamental forms and the embedded Weingarten map of the sequence of spline rings. The details of the setup can be found in [9] or [4].

We consider a subdivision surface $\mathbf{X}$ composed of spline rings

$$\mathbf{x}^m := B\mathbf{P}^m, \quad \mathbf{P}^m := A^m\mathbf{P},$$

where B is a row vector of basis functions forming a partition of unity, $\mathbf{P}$ is a column vector of initial control points, and A is a square *subdivision matrix* with rows summing to one. We assume a standard scheme, i.e., the eigenvalues of A are

$$\lambda_0 = 1, \quad \lambda := \lambda_1 = \lambda_2, \quad \mu := \lambda_3 = \cdots = \lambda_r,$$

where

$$1 > \lambda > \mu > \lambda_i, \quad i > r.$$

Denoting the right eigenvector to λ_i by v_i, we define the *eigenfunctions*

$$f_i := Bv_i$$

and the *eigencoefficients* $\mathbf{q}_i$ by the expansion

$$\mathbf{P} = \sum_i v_i\mathbf{q}_i.$$

Now, we obtain

$$\mathbf{x}^m = \sum_i \lambda_i^m f_i\mathbf{q}_i.$$

Since $f_0 = 1$, we obtain

$$\mathbf{x}^m = \mathbf{q}_0 + \lambda^m [f_1, f_2]\mathbf{Q} + o(\lambda^m), \quad \mathbf{Q} = \begin{bmatrix}\mathbf{q}_1 \\ \mathbf{q}_2\end{bmatrix}.$$

The planar spline ring

$$\boldsymbol{\Psi} := [f_1, f_2]$$

is called the *characteristic map* of the scheme [6], and it is well known that $\mathbf{X}$ is a C_1^k-surface for almost all initial data if $\boldsymbol{\Psi}$ is regular and injective. In particular, the *central normal vector* is

$$\mathbf{n}^{\mathrm{c}} := \lim_{m\to\infty} \frac{{}^{\times}D\mathbf{x}^m}{|{}^{\times}D\mathbf{x}^m|} = \frac{\mathbf{q}_1 \times \mathbf{q}_2}{|\mathbf{q}_1 \times \mathbf{q}_2|},$$

provided that $\mathbf{q}_1, \mathbf{q}_2$ are linearly independent. In the following, we will assume *generic initial data*, i.e.,

$$\det[\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k] \neq 0, \quad i \neq j \neq k \neq i,$$

and note that the set of non-generic data has measure zero so that it can be excluded from our analysis without an essential loss of generality. In particular, $\mathbf{q}_1, \mathbf{q}_2$ are linearly independent for generic initial data so that the central normal is well defined.

To efficiently deal with asymptotic expansions, we introduce an equivalence relation for sequences of functions with coinciding leading terms. We write

$$a^m \overset{c^m}{=} b^m \quad \text{iff} \quad a^m - b^m = o(c^m),$$

where $o(c^m)/c^m$ converges uniformly to zero as $m \to \infty$. For example, $a^m \overset{1}{=} a$ means that a^m converges to a. For vector-valued expressions, the equivalence relation is understood component-wise. For simplicity, $\overset{c^m}{=}$ is mostly replaced by the symbol $\doteq$ with the understanding that the dot refers to the lowest order term specified explicitly on the right hand side of a relation. Hence, the expansion of the sequence of spline rings above now simply reads $\mathbf{x}^m \doteq \mathbf{q}_0 + \lambda^m \boldsymbol{\Psi}\mathbf{Q}$, meaning that the omitted remainder term decays faster than λ^m.

For the following analysis of curvature, we assume for the sake of simplicity (but without loss of generality) that Cartesian coordinates are chosen such that $\mathbf{q}_0 = \mathbf{0}$ is the origin and $\mathbf{n}^{\mathrm{c}} = \mathbf{e}_3$ is the third unit vector. Hence, the eigencoefficients $\mathbf{q}_1, \mathbf{q}_2$ lie in the xy-plane, and there exists a (2×2)-matrix L with

$$\mathbf{Q} = L\mathbf{T}, \quad \mathbf{T} := \begin{bmatrix}1\,0\,0 \\ 0\,1\,0\end{bmatrix}.$$

For generic initial data, $\det L \neq 0$. With $\tilde{\boldsymbol{\Psi}} := \boldsymbol{\Psi} L$ and

$$\varphi := \sum_{\ell=3}^{r} f_\ell \, \mathbf{q}_\ell \cdot \mathbf{n}^{\mathrm{c}},$$

the asymptotic expansion of $\mathbf{x}^m$ involving also the behavior of the normal component reads

$$\mathbf{x}^m \doteq [\lambda^m \tilde{\boldsymbol{\Psi}}, \mu^m \varphi] = \tilde{\mathbf{x}} \operatorname{diag}([\lambda^m, \lambda^m, \mu^m]).$$

The spline ring $\tilde{\mathbf{x}}$ is called the *central surface* of $\mathbf{X}$, and many shape properties of $\mathbf{X}$ are related to those of $\tilde{\mathbf{x}}$, see [4,3].

From [4], we recall without proof the following expansions:

Theorem 5. *The asymptotic expansions of the partial derivatives and the fundamental forms of the spline rings* $\mathbf{x}^m$ *are*

$$\begin{aligned} D\mathbf{x}^m &\doteq \lambda^m D\tilde{\boldsymbol{\Psi}}\, \mathbf{T} = \lambda^m D\boldsymbol{\Psi} L \mathbf{T} \\ I^m &\doteq \lambda^{2m} I, \quad I^{-1} \doteq \lambda^{-2m} I^{-1}, \quad I := D\tilde{\boldsymbol{\Psi}} \cdot D\tilde{\boldsymbol{\Psi}} \\ II^m &\doteq \mu^m II, \quad II := \sqrt{\frac{\det \tilde{I}}{\det I}}\, \tilde{II}, \end{aligned}$$

where $\tilde{I}$ *and* $\tilde{II}$ *are the fundamental forms of the central surface* $\tilde{\mathbf{x}}$*.*

In particular, one easily concludes that the sign of the Gaussian curvature of the spline rings and the central surface are asymptotically equal at non-parabolic points,

$$\operatorname{sign} K_g^m = \operatorname{sign} \det II^m \doteq \operatorname{sign} \det \tilde{II} \quad \text{if} \quad \det \tilde{II} \neq 0.$$

Now we are prepared to derive the asymptotic expansion of the embedded Weingarten map of the spline rings.

Theorem 6. *The embedded Weingarten maps* $\mathbf{W}^m$ *of the spline rings* $\tilde{\mathbf{x}}$ *satisfy*

$$\mathbf{W}^m \doteq (\mu/\lambda^2)^m \begin{bmatrix} V & 0 \\ 0 & 0 \end{bmatrix}, \quad V := D\tilde{\boldsymbol{\Psi}}^{-1} II \cdot D\tilde{\boldsymbol{\Psi}}^{-1}.$$

Proof. By Theorem 5, we obtain for the pseudo-inverse of $D\mathbf{x}^m$

$$(D\mathbf{x}^m)^+ = (D\mathbf{x}^m)^{\mathrm{t}} (I^m)^{-1} \doteq \lambda^{-m}\, \mathbf{T}^{\mathrm{t}} D\tilde{\boldsymbol{\Psi}}^{\mathrm{t}} I^{-1} = \lambda^{-m}\, \mathbf{T}^{\mathrm{t}} D\tilde{\boldsymbol{\Psi}}^{-1}.$$

Hence,

$$\begin{aligned} \mathbf{W}^m &= (D\mathbf{x}^m)^+ II^m \cdot (D\mathbf{x}^m)^+ \doteq (\mu/\lambda^2)^m\, \mathbf{T}^{\mathrm{t}} \big(D\tilde{\boldsymbol{\Psi}}^{-1} II \cdot D\tilde{\boldsymbol{\Psi}}^{-1}\big) \mathbf{T} \\ &= (\mu/\lambda^2)^m\, \mathbf{T}^{\mathrm{t}} V \mathbf{T}, \end{aligned}$$

as stated.

4 Conditions for Curvature Continuity

By Theorem 4, a subdivision surface is C_2^k if and only if $k \geq 2$, and if

$$\mathbf{W}^{\mathrm{c}} := \lim_{m \to \infty} \mathbf{W}^m$$

exists and is constant. Obviously, a limit can exist only in the following cases:

1. $\mu/\lambda^2 > 1$ and $\mathbf{W} = 0$. Although it is often taken for granted that the first condition alone implies divergence of the principal curvatures, it has to be excluded that the second condition cannot be satisfied for generic initial data. This part of the argument, which is not completely trivial, is rarely carried out in detail. The condition $\mathbf{W} = 0$ implies $I\!I = \tilde{I\!I} = 0$ since $D\tilde{\boldsymbol{\Psi}} = D\boldsymbol{\Psi}\, L$ has full rank. Let us consider the central surface $\tilde{\mathbf{x}} = [\tilde{\boldsymbol{\Psi}}, \varphi]$. Its second fundamental form vanishes if and only if its trace is part of a plane. Hence, there exists a vector $\tilde{\mathbf{n}}$ with unit length and a number $\tilde{c}$ such that

$$[1,\, \tilde{\mathbf{x}}] \cdot [\tilde{c},\, \tilde{\mathbf{n}}] = B[e,\, \tilde{v}_1,\, \tilde{v}_2,\, w] \cdot [\tilde{c},\, \tilde{\mathbf{n}}] = 0,$$

   where $e = v_0 = [1; \ldots; 1]$ are the control points of the 1-function, $[\tilde{v}_1, \tilde{v}_2] = [v_1, v_2]L$ are the control points of $\tilde{\boldsymbol{\Psi}}$, and

$$w := \sum_{\ell=3}^{r} v_\ell\, \mathbf{q}_\ell \cdot \mathbf{n}^{\mathrm{c}}$$

   are the control points of φ. Since the functions in B are assumed to form a basis[1], we have

$$[e,\, \tilde{v}_1,\, \tilde{v}_2,\, w] \cdot [\tilde{c},\, \tilde{\mathbf{n}}] = 0.$$

   For generic initial data, $\mathbf{q}_\ell \cdot \mathbf{n}^{\mathrm{c}} \neq 0$ for all $\ell \in \{3, \ldots, r\}$. Hence, linear independence of the eigenvectors $v_0, \ldots, v_r$ implies that the third component $\tilde{\mathbf{n}}_3$ of $\tilde{\mathbf{n}}$ must vanish. This, however, contradicts

$$|\tilde{\mathbf{n}}_3| = \frac{|\det D\tilde{\boldsymbol{\Psi}}|}{|\tilde{\mathbf{x}}_s \times \tilde{\mathbf{x}}_t|} = \frac{|\det D\boldsymbol{\Psi} \cdot \det L|}{|\tilde{\mathbf{x}}_s \times \tilde{\mathbf{x}}_t|} \neq 0.$$

   This verifies in a rigorous way that the case $\mu/\lambda^2 > 1$ cannot yield curvature continuity for generic initial data.
2. $\mu/\lambda^2 < 1$. This case yields curvature continuity with $\mathbf{W}^{\mathrm{c}} = 0$. However, the enforced flat spot at the central point is not acceptable in most applications.
3. $\mu/\lambda^2 = 1$. This is the well known case of bounded curvature, and one can show that $\mathbf{W}$ is constant if and only if

$$f_\ell \in \{f_1^2, f_1 f_2, f_2^2\}, \quad \ell = 3, \ldots, r,$$

   i.e., if the subsub-dominant eigenfunctions are quadratic polynomials in the subdominant ones.

Recently, a new class of C_2^2-schemes was devised by Karciauskas and Peters [1,2] under the label of *guided subdivision*. Generalizing the idea of TURBS [8] in a striking way, these subdivision schemes combine for the first time analytic

[1] If the functions in B are not linearly independent, the argument given here is not valid unless the matrix A is modified in such a way that *ineffective eigenvectors* are removed [9].

smoothness with geometric fairness and ease of use. Still, they are more complicated than standard schemes, like the algorithms of Catmull-Clark or Loop, and in particular, according to the results in [7], the polynomial bi-degree of the patches is at least $(6, 6)$ if the setup described here is used. Hence, curvature discontinuous schemes are not yet obsolete, and their shape properties should be further investigated and optimized. As a result of this paper, the embedded Weingarten limit map, given by the symmetric (2×2) matrix V, is the appropriate tool to do that.

References

1. Karciauskas, K., Peters, J.: Guided Subdivision. Technical report (2005)
2. Karciauskas, K., Peters, J.: Concentric Tesselation Maps and Curvature Continuous Guided Surfaces. Computer Aided Geometric Design 24, 99–111 (2007)
3. Karciauskas, K., Peters, J., Reif, U.: Shape Characterization of Subdivision Surfaces—Case Studies. Computer Aided Geometric Design 21, 601–614 (2004)
4. Peters, J., Reif, U.: Shape Characterization of Subdivision Surfaces—Basic Principles. Computer Aided Geometric Design 21, 585–599 (2004)
5. Pottmann, H.: Industrial Geometry. Unpublished manuscript (2005)
6. Reif, U.: A Unified Approach to Subdivision Algorithms Near Extraordinary Vertices. Computer Aided Geometric Design 12, 153–174 (1995)
7. Reif, U.: A Degree Estimate for Subdivision Surfaces of Higher Regularity. Proc. of the AMS 124(7), 2167–2174 (1996)
8. Reif, U.: TURBS—Topologically Unrestricted Rational B-Splines. Constructive Approximation 14, 57–77 (1998)
9. Reif, U., Peters, J.: Structural Analysis of Subdivision Surfaces — a Summary. In: Jetter, K., et al. (eds.) Topics in Multivariate Approximation and Interpolation, pp. 149–190 (2006)

Curvature-Based Surface Regeneration

Sebastian T. Robinson and Glen Mullineux

Innovative Manufacturing Research Centre, Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK
G.Mullineux@bath.ac.uk

Abstract. An interactive technique for integrating surface inspection with fairing is proposed and demonstrated for quadrilateral meshes. This technique works by creating discrete curvature tensors for each mesh point, interpolating this data to create a finer mesh for inspection, allowing the user to change any curvature information and then regenerating the mesh using the new information.

1 Introduction

Fair or 'A-class' surfaces are a requirement in many design applications. However deciding what is actually meant by a fair curve or surface is a difficult task. Often the definition of what one is is highly dependent on the area concenred and so varies from application to application. Although the precise meaning of what fairness means is still a matter of debate, several popular definitions exist including the following: what the designer expects; what the customer or manager expects; a degree of 'beauty'.

Curves are inherently simpler geometric objects than surface and so there is greater scope for definitions or characterisation of fairness which relate directly to geometric properties. In particular, the following have been proposed.

- "A curve is fair if its curvature plot is continuous and consists of only a few monotone pieces" [1].
- A curve which exhibits curvature plots and tangent plots agreeable to the designer [2].

The natural things to do for surfaces is to try to generalise the equivalent concepts for curves. Accordingly, for surfaces, proposed definitions of fairness include the following.

- A surface whose isophotes (an equivalent to the effect of car showroom strip lighting) are in the form of fair curves is accepted in the automotive industry as being a fair surface [1].
- A surface which exhibits a smooth curvature plot. This can be Gaussian, mean, absolute or even two separate plots of the principal curvatures.
- A surface of minimal energy: this can be bending energy, elastic energy or both [3].

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 378–390, 2007.
© Springer-Verlag Berlin Heidelberg 2007

– If fair curves pass through a surface then that surface can itself be interpreted as being fair; clearly however the choice of curves is critical.
– "... a surface is 'fair' if it has optimal monotonicity and distribution of curvature" [4].

A distinction also can be made between faring to remove noise and fairing to remove undesirable features. Although almost all noise tends towards the creation of undesirable features, it should be noted that not all undesirable features are due to noise and it is sometimes better to combat the two separately. Many sources in the literature make no distinction between these two types of fairing and it is arguable whether a distinction really needs to be made in many applications. Generally if a distinction is made then the removal of noise is referred to as smoothing or filtering and the removal of undesirable features is referred to as fairing or beautification. The technique discussed in this paper is primarily a tool for beautification.

A variety of surface fairing algorithms exist with different levels of automation, but the biggest hurdle in all the fairing algorithms seems to be the ability to smooth a surface while retaining design features. Totally automated fairing processes are always going to have difficulty achieving this, though the anisotropic diffusion methods have produced some impressive results [5]. For the majority of quality control methods some user interaction is required. Farin and Sapidis [6] put this well: "The goal of automatic shape control will never be accomplished completely, but in view of steadily increasing automation in the design process, coming at least as close to the goal as possible seems mandatory... No computer system should attempt to produce 'optimally aesthetic' objects. The finishing touches should always be left to the designer."

With this in mind and the current trend of many recently published fairing techniques to demonstrate the quality of fairing using curvature plots it seems sensible to allow the user to interrogate a surface's curvature plot and suggest better curvature information as input for a fairing algorithm. This heuristic was recently implemented for curves [7], where a planar curve can be approximately reconstructed after its curvature data has been improved by user intervention.

This paper looks at the question of whether the extension of curve reconstruction from curvature data can be extended to surfaces. That is, can a surface be recreated from a knowledge of the 'profile' of its curvature tensor. If the answer is "yes" then it opens up the possibility of fairing a surface by improving its curvature tensor. Various tools for refining the curvature information are possible (though not discussed in depth here) including: manual adjustment, Fourier analysis, and the use of continuous surface representations of the curvature data. The use of curvature Bézier control points would doubtless be a useful tool in industry [8].

If recreation is possible, then similarly a surface can be recreated from point information sampled from it, together with information about the curvature tensor at these points. Such curvature data can of course be estimated approximately from the points themselves. This opens up applications in reverse engineering of sculptured surfaces. It also allows impinges on the area of product design and

styling. Here designers often prefer to work with physical objects and these can be created from a CAD model using rapid prototyping techniques. If a designer then modifies the physical prototype, it can be recaptured approximately as a CAD model by suitable scanning [9]. The ability to recreate (and improve) a surface determined by sampled points can assist in this process.

In this paper, a fairing method is proposed which takes a mesh surface, generates an initial curvature plot for it, allows the user to suggest a better curvature plot and finally grows a new surface between the fixed pre-defined boundaries of the surface based on the new curvature information. Alternative techniques are possible. For example, one approach is to use a functional which describes an appropriate geometric attribute of the proposed surface (such as curvature) and then seek a surface solution which minimises that functional [10].

Section 2 gives the basic approach used for recreating a surface from its curvature profile. The case of essentially quadrilateral meshes is discussed. In addition, triangular meshes have been implemented, tested and found to work satisfactorily. It is anticipated that the approach can be extended to arbitrary meshes. Section 3 gives an example of applying the approach. The original mesh used for this is taken from published data sampled from a ship's hull. It is seen that the surface can be recreated and that manual adjustment of the curvature profile allows surface improvements to be undertaken. Finally Section 4 presents some conclusions.

2 Methodology

For simplicity the case of the four sided mesh is discussed in this paper. A 2D parameterisation of the surface is required for the purposes of interpolating a finer mesh and for the growing phase of the algorithm where parametric directions are required. In the case of a four sided surface with uniform rectangular sampling, a parameterisation is trivial.

The basic approach operates using the following stages.

- Obtain the original mesh of points
- Parameterise over 2D space
- Calculate the discrete curvature tensors
- Refine the mesh density if required
- Adjust the curvature information
- Regenerate the mesh from the boundaries using the new curvature data
- Repeat if necessary

For the mesh information, together with a suitable parameterisation, it is necessary to obtain curvature tensor information at the individual points (cf. Figure 1). At each point the curvature tensor information includes the following.

- Unit normal vector
- Vector in the direction of maximal curvature
- Vector in the direction of minimal curvature

– Scalar value of maximal curvature
– Scalar value of minimal curvature

The scalar values are stored separately from the curvature vectors to allow for positive and negative curvature values. The three vectors are orthogonal to each other and because of the separate storage of the scalar values the curvature vectors are free to point in either direction.

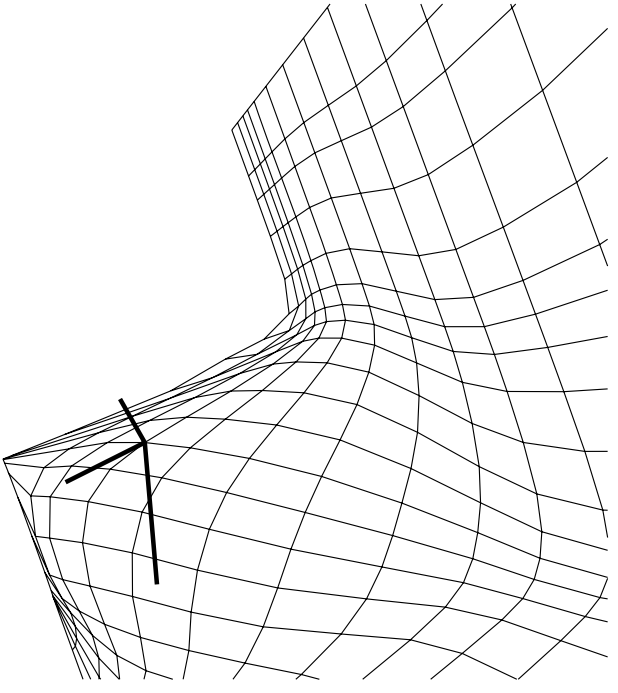


Fig. 1. Curvature tensor on a quadrilateral mesh

The curvature tensors are calculated using a circle fitting method [11]. In this, the point being considered and two of its neighbours in various combinations of the parametric directions are considered. In the case of the quadrilateral mesh used, the combinations of three points shown in Figure 2 are used.

A circle is fitted to each set of three points to represent a curve passing through them. The osculating radius of this circle corresponds to an approximate curvature of the curve. By applying Meusnier's theorem [12] using the angle between the plane of the circle and the surface normal, the normal curvature of the curve through the surface can be approximated. Using a selection of these calculated normal curvatures combined with the angles at which each circle plane cuts the surface tangent plane, values for the maximal and minimal curvature can be calculated through Euler's theorem [12] and least squares fitting.

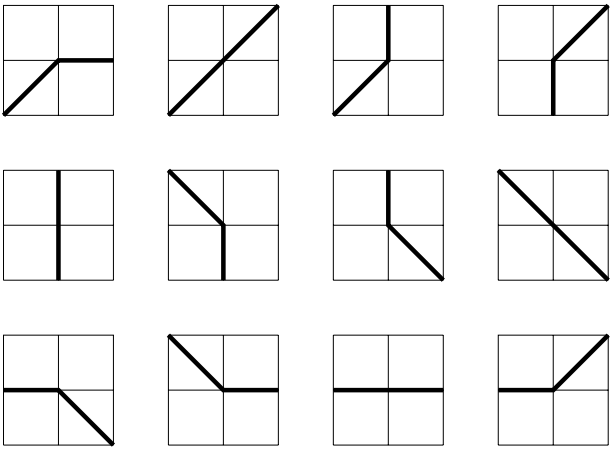


Fig. 2. Combinations of three points

It has been the authors' experience that using points which veer around more than 60° from a straight line in the surface tangent plane leads to poor curvature estimations.

The next stage is to refine the mesh to one that is denser. This enables the mesh to be more continuous and facilitates more specific control over the curvature.

As the mesh is regenerated after this stage, the locations of the new points are not too important and are simply parametric interpolations of the surrounding points. What is important is the curvature information. This requires the applications of appropraite transforms and here this is achieved using representation in terms of quaternions [13]. What is required is the ability to transform a tensor to positions between two given points and in this way form an interpolation. This is done by applying a series of 'SLerps' (spherical linear interpolations) [14]. Suppose the two position are achieved by applying two transforms (to a given reference frame) represented by quaternions q_1 and q_2. Then the following quaternion is constructed:

$$\text{qSLerp}(t, q_1, q_2) \quad = \quad q_1(q_1^{-1}q_2)^t$$

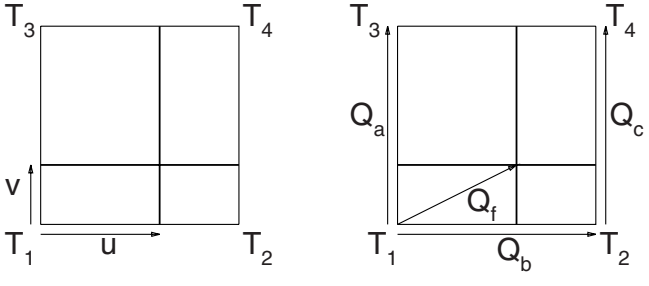


Fig. 3. Interpolated direction of new tensor

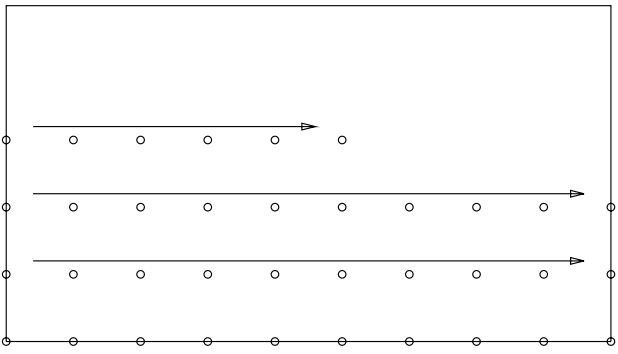


Fig. 4. Growing the new surface

where t is a scalar parameter with $0 \leq t \leq 1$. Clearly, when $t = 0$, this new quaternion is simply q_1; and it is q_2 when $t = 1$. So as t varies across values in between, interpolated transforms are achieved.

Now consider the case shown in Figure 3. Here T_1, T_2, T_3 and T_4 are the tensors from the original mesh which surround the new point, whose coordinates are the parametric values u and v lying between 0 and 1. Along one edge, Q_a is the quaternion which transforms the direction of T_1 to the direction of T_3, along an ajoining edge, Q_b is the quaternion which transforms the direction of T_1 to the direction of T_2, and, along a third edge, Q_c is the quaternion which transforms the direction of T_2 to the direction of T_4. Finally, Q_f is the quaternion needed to transform T_1 to the tensor at the new point. This is obtained through SLerp transforms as follows:

$$Q_f = \text{qSLerp}(u,\ \text{qSLerp}(v, 1, Q_a),\ Q_b\text{qSLerp}(v, 1, Q_c)\ ),$$

where 1 is the unit scalar quaternion.

The calculation of directions is most simply achieved through application of SLerp transforms to unit length versions of the original tensors. Special care must be taken to ensure the tensors do not over-rotate due to the principle curvatures of neighbouring points being represented in close to opposite directions. Once the new position has been established the scalar values of maximal and minimal curvature are calculated using the same barycentric weightings used for the location calculations. These values are then reapplied to the newly directed tensors.

Whether the mesh has been refined of not, the curvature plot is now displayed to the user. The user has the option to change any information about the curvature at each tensor. It should be kept in mind that the mesh still needs to meet the boundary conditions so the requested curvature information given by the user is almost always compromised to meet this and any conflicting curvature conditions.

In the case of the four-sided surface, four individual surfaces are grown, one starting from each side and the four surfaces are then blended. Each individual surface starts along one side, where the information is already available. It is then built up along the next rows of points wthin in the mesh, proceeding from

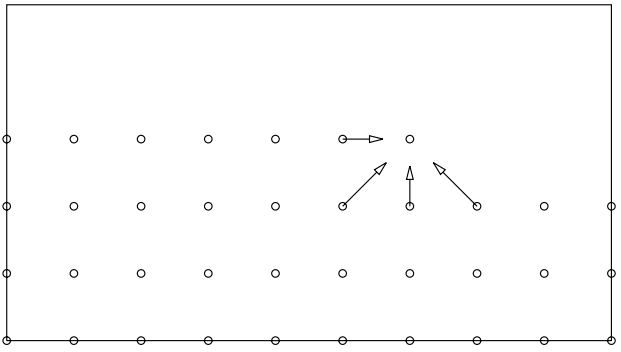


Fig. 5. Mask of which neighbours are used to estimate the new position

from end of the row to the other. The next row is then tackled, and so on. Figure 4 illustrates this idea showing the state when two rows have been dealt with and the third is in progress.

In proceeding along each row to construct the surface, for each new point there are four neighbouring points which have already been processed and for which surface information is available to construct the location for the new point. These four points are shown in Figure 5. Each of the four is used to estimate the new location and the four estimates are averaged. Each of the individual estimates is obtained as follows.

For a given new point and one of it neighbours, the neighbour's tensor is used to project an initial estimate of the location of the new point. Then an initial line is constructed joining this estimated location to the neighbour. Suppose this has length D. The tensors at the two points are both known. The joining line is projected onto the surface tangent plane determined by each of the tensors separately. The angle between these projected lines and the vectors of maximal curvature can then be used alongside the scalar values of curvature to find two normal curvatures, one for each tensor, in the direction of the line joining the two points, using Euler's theorem (cf. Figure 6). These two values of curvature are averaged and then inverted to create an approximate value R of the osculating radius for a curve linking the two points.

The location of the new point is now re-estimated. A circular arc is constructed starting at the neighbour and with radius R lying in the plane defined by the surface normal $\mathbf{N}$ at the neighbour and the unit vector $\mathbf{L}$ along the projection of the joining line onto the tangent plane at the neightbour. The new estimate is that point along this arc which is distance D away from the neighbour as in Figure 7. The position vector of this estimate relative to the neighbour is $\mathbf{P}$ where

$$\mathbf{P} \;=\; (\cos\theta)\mathbf{L} \;+\; (\sin\theta)\mathbf{N}$$

and

$$\theta \;=\; \sin^{-1}\left(\frac{D}{2R}\right).$$

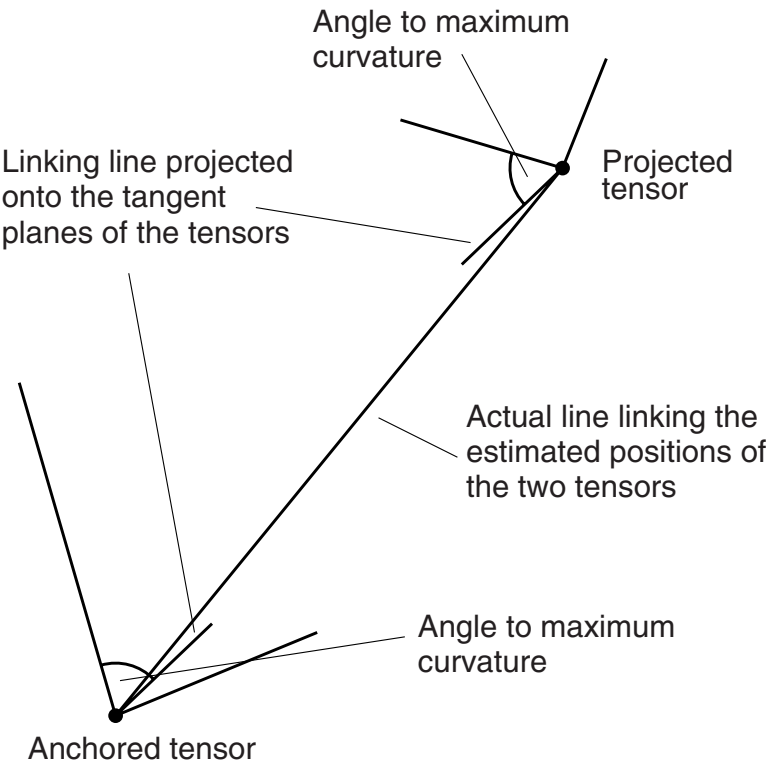


Fig. 6. New point estimation

At this point it is worth noting that the curvature values used (either as calculated values or user suggested values) must not be so high as to suggest osculating circles with a diameter near to or less than the projecting distance as the projection will be rendered inaccurate or (if the the diameter is less than the projection distance) impossible.

Once a surface has been grown from each of the four sides, a final surface is blended using the following weights for each mesh point at any given a parametric position (u, v) as in Figure 8.

$w_1 = (u-1)/w_{tot}$	the surface grown from the left hand side
$w_2 = (u)/w_{tot}$	the surface grown from the right hand side
$w_3 = (v-1)/w_{tot}$	the surface grown from the lower side
$w_4 = (v)/w_{tot}$	the surface grown from the upper side

where $w_{tot} = w_1 + w_2 + w_3 + w_4$.

Naturally, the entire process can then be repeated if the results are unsatisfactory.

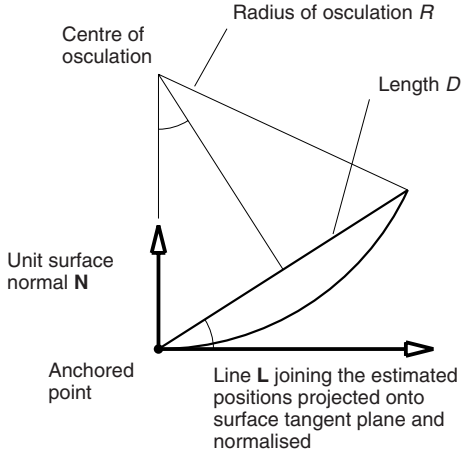


Fig. 7. Location of new point

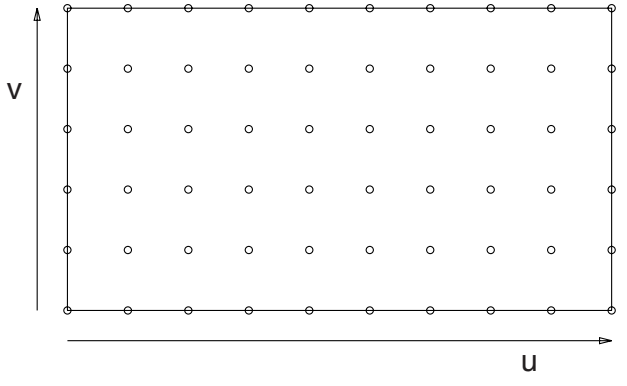


Fig. 8. Parameterisation of surface

3 Example

The example used is a CAD model of a container ship [15] with a bulbous bow with a rectangular sampling. Only half the hull is needed as the model is symmetrical. Also the hull has been shortened by a factor of seven for ease of viewing. Three views of the hull as used here are shown in Figure 9. The mesh data is used to estimate the curvature tensor at each point and Figure 10 shows the corresponding directions and magnitudes. These tensors and their locations can be refined and the result is seen in Figure 11.

The regenerated hull, even without any user adjustment of the curvature data, does not recreate the original surface perfectly, though it does follow

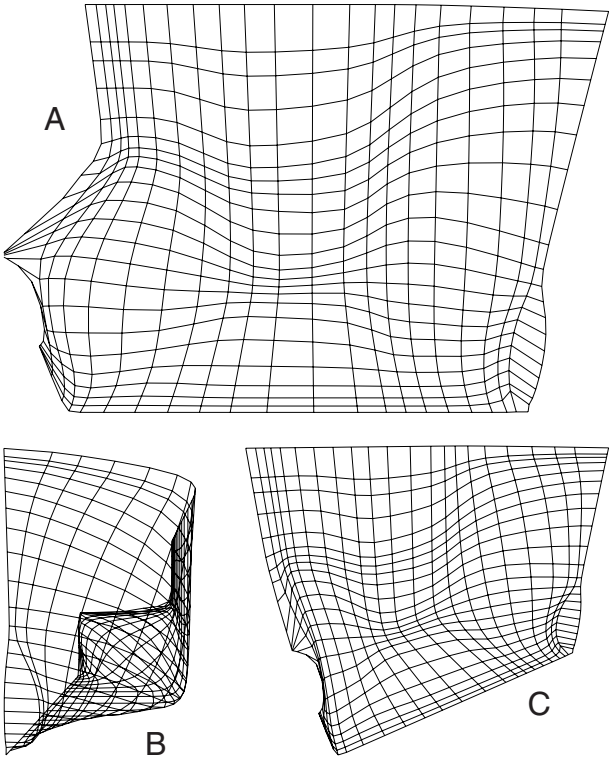


Fig. 9. Original hull data: (A) General view from high angle, (B) Front view, (C) View from inside

the original shape very closely. Inherently the process tends to smooth surfaces when the mesh density is increased, even if the curvature is not adjusted. In the case where the curvature is not adjusted by user intervention, all the features of the original surface are faithfully recreated including any noise (the curvature smoothing is where the noise removal occurs). The regenerated surface always tends to remain just within the convex hull of the original surface unless new curvature data is given which forces the surface to depart significantly from the original shape. The newly grown mesh is shown in Figure 12.

4 Conclusions

A method for recreating a surface from the profile of its curvature tensor has been proposed and shown to work satisfactorily. The curvature tensor is specified at points within a quadrilateral mesh, and indeed can be estimated from the mesh itself using finite difference approximations. This leads into an interactive method of surface inspection and fairing by trying to 'improve' the curvature

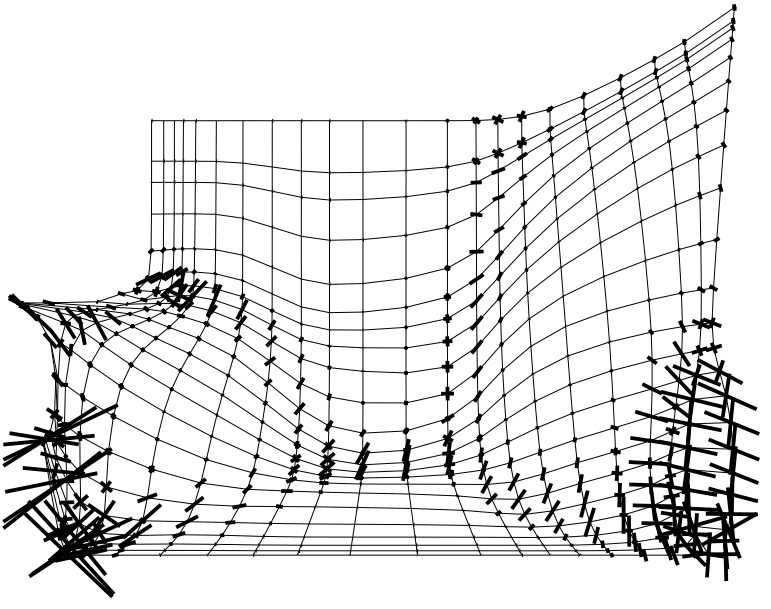


Fig. 10. Discrete curvature tensors on original hull

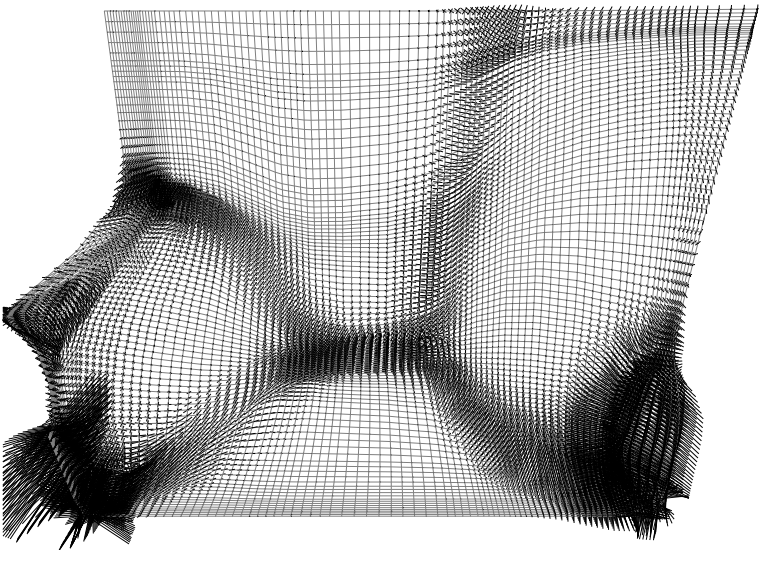


Fig. 11. Discrete curvature tensors on refined mesh of hull

profile either manually or automatically. With suitable and intuitive tools for user curvature adjustment, it is anticipated that this can prove to be a useful tool for the creation of A-class surfaces.

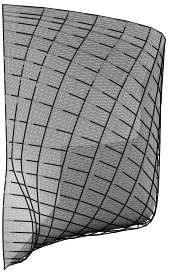


Fig. 12. Newly grown mesh is much smoother although it does not perfectly match original surface as seen in lower right corner

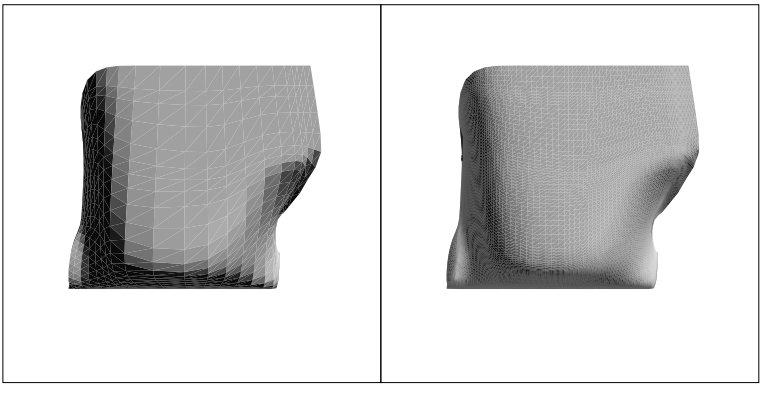


Fig. 13. Original and refined hull

The case of quadrilateral meshes only has been covered. A similar method for dealing with very simple triangular meshes has in fact also been successfully implemented. The main hurdle for taking this technique up to more complicated meshes seems to lie in the adoption of suitable parameterisations and associated growing techniques and directions.

Acknowledgement

The first named author is funded by an EPSRC DTA studentship provided via the IMRC at the University of Bath. This support is gratefully acknowledged.

References

1. Farin, G.: Curves and Surfaces for CAGD—A Practical Guide, 5th edn. Academic Press, San Diego (2002)
2. Renz, W.: Interactive Smoothing of Digitized Point Data. Computer-Aided Design 14, 267–269 (1982)

3. Miao, Y., Shou, H., Feng, J., Peng, Q., Forrest, A.R.: Bézier Surfaces of Minimal Internal Energy. The Mathematics of Surfaces XI, pp. 161–183 (2005)
4. Peters, J.: Smoothness, Fairness and the Need for Better Multi-sided Patches. Contemporary Mathematics 334, 55–64 (2003)
5. Clarenc, U., Diewald, U., Rumpf, M.: Anisotropic Geometric Diffusion in Surface Processing. In: Proc. Visualisation 2000, pp. 397–405 (2000)
6. Farin, G., Sapidis, N.: Fairing Curves. IEEE Computer Graphics & Applications 9, 52–57 (1989)
7. Mullineux, G., Robinson, S.T.: Fairing Point Sets using Curvature. Computer-Aided Design 39, 27–34 (2007)
8. Albat, F., Muller, R.: Free-form Surface Construction in a Commercial CAD/CAM System. In: The Mathematics of Surfaces X, pp. 1–13. Springer, Heidelberg (2005)
9. Prieto, P.A., Wright, D.K.: A Novel Method for Early Formal Developments using Computer Aided Design and Rapid Prototyping Technology. In: Proc. Instn. Mech. Engrs. Part B: Journal of Engineering Manufacture, 217, 695–698 (2003)
10. Bloor, M.I.G., Wilson, M.J., Hagen, H.: The Smoothing Properties of Variational Schemes for Surface Design. Computer Aided Geometric Design 12, 381–394 (1995)
11. Goodman, T., Martin, R.R. (eds.): Estimation of curvatures from point data. The Mathematics of Surfaces VII, Information Geometers, pp. 131–144 (1997)
12. Pressley, A.: Elementary Differential Geometry. Springer, London (2001)
13. Röschel, O.: Rational motion design - a survey. Computer-Aided Design 30, 169–178 (1998)
14. Shoemake, K.: Animating rotation with quaternion curves. In: Proc. 12th Annual Conference on Computer Graphics and Interactive Techniques, pp. 245–254 (1985)
15. Anon.: FSSBROCH bulbous bowed ship [last amended, (1996)], [last accessed January 2007] [online]. Available from
`http://www.dataship.com/vmrl/fssbroch_2.wrl`

Bounded Curvature Subdivision Without Eigenanalysis

Malcolm A. Sabin[1], Thomas J. Cashman[2],
Ursula H. Augsdorfer[2], and Neil A. Dodgson[2]

[1] Numerical Geometry Ltd., UK
malcolm@geometry.demon.co.uk
[2] Computer Laboratory, University of Cambridge, CB3 0FD, UK

Abstract. It has long been known how to achieve bounded curvature at extraordinary points of a subdivision scheme by using eigenanalysis and then adjusting the mask of each extraordinary point. This paper provides an alternative insight, based on the use of second divided differences, and applies it to three familiar schemes. A single concept is shown to work in three different contexts. In each case a bounded curvature variant results, with a very simple and elegant implementation.

1 Background

The earliest surface subdivision schemes suffered from the problem that at extraordinary vertices, where the number of faces meeting in the control polygon was different from the regular number, the curvature was not continuous. Curvature continuity could hardly be expected from the generalisation of the biquadratic B-spline [3,4], but in the case of the generalisations of the bicubic B-spline [3] and the 3- and 4-direction box-splines [10,15] the regular regions have continuous second derivatives and it was disappointing that this property was not maintained at extraordinary vertices.

It was soon realised that the misbehaviour was not simply a step change in 2nd derivative. It was that estimators of curvature based on finite numbers of refinements diverged as the number of refinements increased. The problem was not an infinite third derivative, but an infinite second derivative! There is a proof [13] that the bicubic B-spline cannot have a continuous second derivative at extraordinary vertices, however the coefficients there are manipulated, and better proofs by Reif [12] show that the degree of polynomial patches in regular parts of the mesh would have to be at least bi-6 (or higher if the continuity in the regular parts was higher than two) if continuity of second derivative was to be achieved without forcing the curvatures to zero at such points.

However, it is possible, by tuning the influences of the EV on its neighbouring new vertices, to prevent the divergence of curvature. The misbehaviour is still there, but it takes the form of slight undulations of curvature between upper and lower limits in every neighbourhood of the EV, however small. Further tuning of the outer coefficients of the mask at the extraordinary point [1] can reduce

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 391–411, 2007.
© Springer-Verlag Berlin Heidelberg 2007

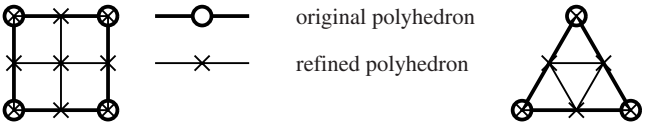


Fig. 1. 1:4 refinement

the size of these variations, though not eliminate them, but the big step is the bounding of the curvature, which can be achieved (apart from a couple of low-valency special cases) purely by tuning the influence of the old extraordinary vertex on its immediate new neighbours and on its successor.

2 New Results

This paper describes a new way of looking at these effects. We find surprising similarities between three of the well known schemes, and present variants of them which have bounded curvature at all extraordinary vertices and which have new vertices inside the convex hull of the old at each stage thus ensuring the variation-diminishing property. We also suggest ways in which the tuning of the outer parts of the mask can be argued without recourse to optimisation.

Dyn, Levin and Gregory [6] explain how, for any given univariate scheme, a new scheme can be set up which expresses the differences after refinement to the differences before. This idea was used [14] to express a univariate scheme in terms of the behaviour of second divided differences. This idea can also be applied in the bivariate context, and Section 4 looks at the differences that actually occur during the multi-stage constructions in terms of which the familiar Catmull-Clark, Loop and 4-8 schemes were originally described. All of them construct edge- and face-vertices by setting a mean second divided difference to zero and vertex vertices so that a mean second divided difference is the same before and after refinement.

In Section 5 we show that this is not coincidence. Such a setting is essential for the bounded curvature property to hold. Unfortunately, the rules of the original schemes use accurate mean second divided differences only when the mesh is affine regular.

In Section 6 we improve these schemes by using better estimators of mean second divided difference, and thereby achieve bounded curvature for all of them.

3 Terminology and Notation

3.1 Kinds of New Vertices

The connectivity of the new polyhedron is strongly related to that of the original. In the primal schemes considered here we think of each old face as being divided into four new faces.

In the quad case there are new vertices associated with the old faces and with the old edges, as well as with the old vertices[1]. We call these **f-vertices**, **e-vertices** and **v-vertices**, respectively. The triangle case has just e-vertices and v-vertices.

3.2 Masks and Stencils

The coefficients in the weighted means used to construct new vertices are conveniently laid out in a pattern echoing the pattern of the vertices. These can be organised either on the basis of what influence an original vertex has on its neighbours *(a mask)*, or on the basis of what influence the neighbouring old vertices have on a given new vertex *(the stencils)*. In the univariate case the mask is the column of the subdivision matrix and the stencils are the rows. Stencils can be used for any calculation of linear combinations of old vertices, and there are therefore appropriate stencils for differences.

3.3 Differences, Derivatives and Curvature

Four closely related concepts are used in this paper.

Differences apply to sequences of numbers or points. The first difference is obtained by subtracting its predecessor from each element in the sequence. The second difference is obtained by subtracting each first difference from the next first difference in sequence. This slight difference in handling means that a second difference is associated with the central vertex of a consecutive triple.

Divided Differences apply to sequences of numbers or points which have associated *abscissa values*. The first divided difference is just the first difference divided by the difference in abscissa values. The significance here is that the abscissa value difference for a refined polygon is half that for the original polygon.

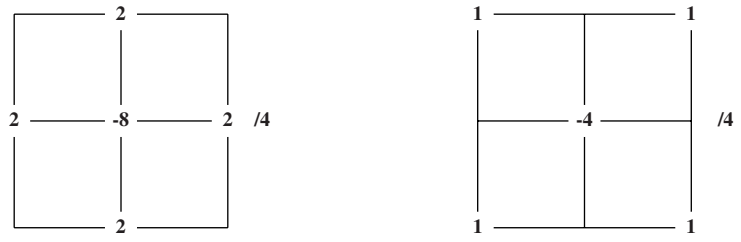
If a given sufficiently smooth function or curve is sampled using successively shorter intervals, the divided difference converges towards the *derivative*, which applies to either functions or parametric curves. The second divided difference (the divided difference of the first divided difference, the abscissa values of the first divided differences being at the centres of the intervals over which they apply) converges towards the second derivative. The derivatives of a curve are vectors, those of a function are scalars.

We do not use first divided differences in this paper, but we shall denote the second divided difference in the x direction by δ_{xx}.

Curvature applies only to curves, not to sequences of points. It is a geometric quantity which is not dependent on the arbitrary assignment of abscissa values. It is equal to the second derivative when arc length is taken as the abscissa.

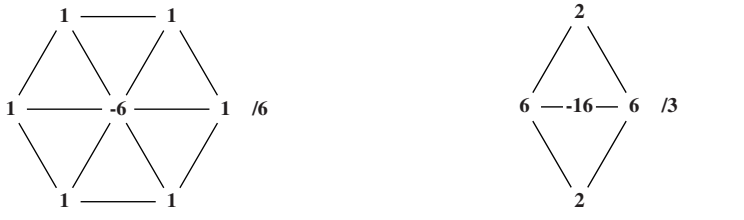
A surface has a *mean curvature* which is a scalar. It is normally taken to be the mean of the two principal curvatures, but in fact is given equally by the

[1] Associated in the loose sense that the new vertex will be positioned somewhere near the centroid of its associate object.



(a) St George's cross stencil (b) St Andrew's cross stencil

Fig. 2. Quad grid stencils for mean second divided difference



(a) Vertex stencil (b) Mid-edge stencil

Fig. 3. Triangle grid stencils for mean second divided difference

mean of the normal curvatures in any two orthogonal directions, or by the mean of normal curvatures in any regularly distributed set of directions.

Mean second divided difference corresponds to mean curvature in the same way that second divided difference corresponds to curvature. It is the mean of the second divided differences taken in two grid-orthogonal directions, or the mean of second divided differences taken in a rotationally symmetric way on a triangle grid. We shall denote this value by m.

We can measure this at a vertex of a polyhedron by using local stencils. In the case of a quad grid the stencil can use either the St.George's cross (grid-aligned) or the St.Andrew's cross (diagonal directions) as shown in Figure 2, or some linear combination of the two. If the quad grid has vertices with x and y on a regular grid, and with z on a quadratic surface, the value of any weighted mean of the two stencils gives the same answer.

In the case of a triangle grid it is conveniently given at the vertices by the mean of the three second divided differences in the grid directions. Figure 3(b) shows a stencil which can be used for new e-vertices in terms of the neighbouring old vertices.

The denominators are chosen to be appropriate for the side of an original facet to be unit abscissa distance, but in fact provided that scales are chosen consistently between before refinement and after, multiplying all divided differences by the same factor does not alter the conclusions which will eventually be drawn.

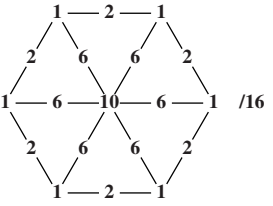
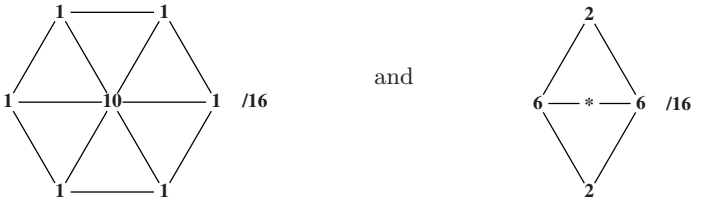


Fig. 4. Loop mask



(a) v-vertex (b) e-vertex

Fig. 5. Loop stencils

We can apply mean second divided difference stencils before refinement, or after. We call these the *original* and *refined* mean second divided differences. It is also possible to apply them partway through a multi-stage refinement step, using a mixture of original and new vertices, and we call the results of this a **mixed** mean second divided difference.

4 At Regular Vertices

An important part of the analysis of subdivision schemes uses the fact that from a given scheme, a *divided difference scheme* can be determined [6], which relates the first divided differences of the refined polyhedron to those of the original. Every scheme which converges has such a difference scheme. Then a second divided difference scheme can be determined in like manner, and every scheme which has a convergent first derivative has such a scheme.

The divided difference scheme is derived from the original by dividing the z-transform of the mask by a $(1 + z)/2$ factor in a particular direction [5].

When we look at some familiar schemes in this light they look very simple and surprisingly similar to each other.

4.1 Loop

The Loop scheme [10] is a generalisation (by permitting extraordinary vertices) of the three-direction boxspline. Its mask in the regular region is shown in Figure 4, and therefore it has stencils for v- and e-vertices as shown in Figure 5.

The subdivision scheme has two stages:

- create new e-vertices at the places defined by the stencil in Figure 5(b).
- create new v-vertices at a suitable weighted mean of the adjacent new e-vertices and the old v-vertex. If the mean of the new e-vertices is $\bar{b}$ and the old v-vertex A, then the new v-vertex, a, is given by $5a = 2A + 3\bar{b}$. We derive these weights from the stencils

$$\begin{aligned} 16b &= 6A + 10\bar{B}, \\ 10\bar{B} &= 16\bar{b} - 6A, \\ 16a &= 10A + 6\bar{B}, \\ &= 10A + 6(16\bar{b} - 6A)/10, \\ 160a &= 100A + 96\bar{b} - 36A, \\ &= 64A + 96\bar{b}, \\ 5a &= 2A + 3\bar{b}. \end{aligned} \tag{1}$$

e-vertices. This construction places the e-vertices where the mixed mean second divided difference is zero.

v-vertices. This construction places the v-vertices so that the mean second divided difference after refinement is equal to the value before.

$$\begin{aligned} 6(a - \bar{b}) &= 6(A - \bar{B})/4, \\ a &= \bar{b} + A/4 - \bar{B}/4, \\ 16\bar{b} &= 6A + 10\bar{B}, \\ \bar{B} &= 16\bar{b}/10 - 6A/10, \\ a &= \bar{b} + A/4 - (16\bar{b}/10 - 6A/10)/4, \\ &= (40\bar{b} + 10A - 16\bar{b} + 6A)/40, \\ &= (24\bar{b} + 16A)/40, \\ &= (3\bar{b} + 2A)/5. \end{aligned} \tag{2}$$

4.2 Catmull-Clark

The Catmull-Clark scheme [3] is a generalisation (by permitting extraordinary vertices) of the two-direction bicubic boxspline. It was originally defined as a three stage process:-

- create new f-vertices at the centroid of the vertices of each face
- create new e-vertices at the centroid of the adjacent old and f-vertices
- create new v-vertices at a suitable weighted mean of the adjacent e- and f-vertices with the old v-vertex.

f-vertices. The construction places the f-vertices at the position where the mixed mean second divided difference is zero.

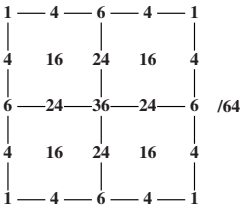


Fig. 6. Catmull-Clark mask

e-vertices. The construction places the e-vertices at the position where the mixed mean second divided difference is zero.

v-vertices. From the box-spline definition we can determine that the mask for the regular case is as shown in Figure 6.

Let the extraordinary vertex be denoted by A, the mean of the edge-connected vertices $\bar{B}$ and the mean of the face-connected vertices $\bar{C}$. Also let the corresponding points after refinement be a, $\bar{b}$ and $\bar{c}$.

From the refinement process we have

$$16 \begin{bmatrix} a \\ \bar{b} \\ \bar{c} \end{bmatrix} = \begin{bmatrix} 9 & 6 & 1 \\ 6 & 8 & 2 \\ 4 & 8 & 4 \end{bmatrix} \begin{bmatrix} A \\ \bar{B} \\ \bar{C} \end{bmatrix}, \tag{3}$$

and thence that

$$16 \begin{bmatrix} \bar{b} - a \\ \bar{c} - a \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} \bar{B} - A \\ \bar{C} - A \end{bmatrix}. \tag{4}$$

so that neither of the two basic mean second divided differences remains invariant. However, if we write

$$\begin{aligned} M_g &= \frac{1}{4}(\bar{B} - A), \\ M_a &= \frac{1}{4}\left(\frac{\bar{C} - A}{2}\right), \\ m_g &= (\bar{b} - a), \\ m_a &= \left(\frac{\bar{c} - a}{2}\right), \end{aligned} \tag{5}$$

we find that

$$m_g + 2m_a = M_g + 2M_a. \tag{6}$$

This is therefore the appropriate combination of the two stencils to consider. It gives as the equation for a:

$$\begin{aligned} \bar{b} + \bar{c} - 2a &= \frac{1}{4}(\bar{B} + \bar{C} - 2A), \\ 8a &= 4\bar{b} + 4\bar{c} + 2A - \bar{B} - \bar{C}. \end{aligned} \tag{7}$$

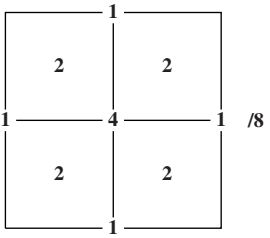


Fig. 7. Mask for 4-8 half-step

As in the Loop case above we can either substitute for $\bar{b}$ and $\bar{c}$ in terms of $\bar{B}$ and $\bar{C}$ to demonstrate the convex hull property, or else substitute for $\bar{B}$ and $\bar{C}$ in terms of $\bar{b}$ and $\bar{c}$ to give an expression in terms of the new local points only.

4.3 4-8

The 4-8 scheme [15], is a generalisation (by permitting extraordinary vertices) of the four-direction boxspline. It is neatly implemented as a $\sqrt{2}$ scheme in which at alternate refinements the grid switches direction between NSEW and diagonal. In this form it is described in terms of a two-stage process:

- create new f-vertices at the centroid of each old face
- create new v-vertices at a suitable weighted mean of the adjacent f-vertices with the old v-vertex.

From the box-spline view we can determine that the $\sqrt{2}$ mask is as shown in Figure 7.

f-vertices. The construction places the f-vertices at the position where the mixed mean second divided difference is zero. In fact the construction is identical to that of the Catmull-Clark f-vertices.

v-vertices. Using the same a, $\bar{b}$, $\bar{C}$ notation as before we have the facts that (from the mask)

$$\begin{aligned} 8a &= 4A + 4\bar{B}, \\ 8\bar{b} &= 2A + 4\bar{B} + 2\bar{C}, \\ 8(\bar{b} - a) &= -2A + 2\bar{C}, \\ \bar{b} - a &= \frac{1}{2}\left(\frac{\bar{C} - A}{2}\right). \end{aligned} \tag{8}$$

Thus the new St.Georges mean second divided difference (which is the only one available) is equally to an appropriately scaled old St.Andrews one.

This mixing of two different stencils is a bit messy, because we cannot prove invariance of anything through a refinement step, and cannot therefore show by a one-line proof any properties of the limit surface. It need not be wrong, because if the grid is regular in x and y and has vertices whose z values are quadratic in x and y, the two stencils give the same answer anyway.

We shall see below why this formulation should be regarded as a coincidence rather than as anything significant.

5 Why?

There is clearly a pattern here. In all three schemes, new f- and e-vertices are being set by making their mixed mean second divided difference zero, and new v-vertices are being set by making the value of a mean second divided difference after refinement equal to that before. It is relatively plausible that new v-vertices should inherit the old mean second difference values; after all, we are looking for the curvature not to change much, and curvature is closely related to second divided differences, but it is far less obvious why the e- and f-vertices should be set using a zero value.

5.1 Why Zero Mean Second Differences for f- and e-Vertices?

The original versions of Loop [10], Catmull-Clark [3] and 4-8 [15] behave well at regular vertices, with bounded (and in fact continuous) curvature there.

Necessary conditions for bounded curvature are that the dominant eigenvalue associated with the $\omega = \pm 2$ Fourier components (μ_2) and the subdominant eigenvalue of the $\omega = 0$ Fourier component (μ_0) about a vertex should be the square of the dominant eigenvalue of the $\omega = \pm 1$ Fourier component (λ), and that all other eigenvalues should be strictly smaller than the two μ values.

In the case of regular vertices these conditions are satisfied in all these schemes.

Because all the other eigenvalues are smaller, we are really focussing on what is happening to second derivatives, and the space of these has only three dimensions. We may take a basis in which two correspond to saddle shapes, and the third to cup shapes. Because of the linearity of the scheme in the control point ordinates, any quadratic shape can be represented as a linear combination of these three components.

The second Fourier component is that which corresponds to saddle-shaped surfaces, and ideally its limit surface should be $z = x^2 - y^2$ or $z = 2xy$ where the xy-plane is the tangent plane at the extraordinary point. If the scheme is to have bounded curvature, then when the old vertices lie on either of these surfaces the new vertices should also. This surface has zero mean second derivative.

Now by symmetry, for pure saddle data there is zero offset: the old control points lie exactly on the limit surface, so that the choice of v-vertices will not influence this case. Any plausible setting will leave the v-vertices at the old vertices, and so setting the mixed second difference to zero is exactly the right thing to do to ensure that the new points also lie on the same surface.

5.2 Why Set the New Mean Second Difference to the Old for v-Vertices?

The simple answer to this question is to say that the overall curvature has to remain much the same as refinement proceeds. However, this answer is shown to be slightly too simple by looking at the 4-8 scheme. To interpret this scheme in terms of second divided differences and setting the new value to equal the old at a vertex requires that different mean second divided differences have to be used for the before and after cases. We equate the St.George's version after to the St.Andrew's before and then use the new St.Andrew's stencil for the next step of refinement.

The real reason why this strategy works for Catmull-Clark and Loop is that in those two cases the stencil (derived from the eigenanalysis) which gives the limit surface mean second derivative is exactly a mean second difference. (In the Catmull Clark case the mean second derivative stencil is a particular linear combination of the St.George's and St.Andrew's crosses.) Thus the mean second difference is exactly what needs to be kept constant to achieve bounded second derivatives in the regular grid case. In the 4-8 case the stencil for the mean second derivative is significantly larger, and it may be coincidence that the condition can be expressed in terms of different smaller stencils being equal.

In the limit, the condition of continuity of curvature demands that the local limit shape should be quadratic. In this case the two estimators give exactly the same value and so any linear combination will give the $\mu_0 = \lambda^2$ property. However, different choices give different schemes. 4-8 cannot use the St.Andrew's cross after refinement because it references other v-vertices, not yet available. The 4-3 scheme equates the St.Andrews cross before and after refinement.

6 At Extraordinary Vertices

This viewpoint now provides a justification for applying the same recipe. *Determine new e- and f-vertices by setting the mixed mean second divided difference to zero, and then v-vertices by setting it in some sense equal to the old.*

The difference is in the choice of stencils for the mean second divided difference. The original definition of second divided difference no longer applies, particularly in the case of odd valencies.

However, we can exploit the symmetry of v-vertices to make a plausible alternative definition. Each of the mean second divided difference stencils for v-vertices in Figures 2 and 3 above is essentially a vector from the central vertex to the centroid of a set of nearby vertices.

For e-vertices the situation is more interesting, but we can still exploit the mirror symmetry of the natural configuration across an edge to give a sensible formulation. *The key idea is that we are using the natural configuration to give abscissae which can then be used to determine second divided differences.*

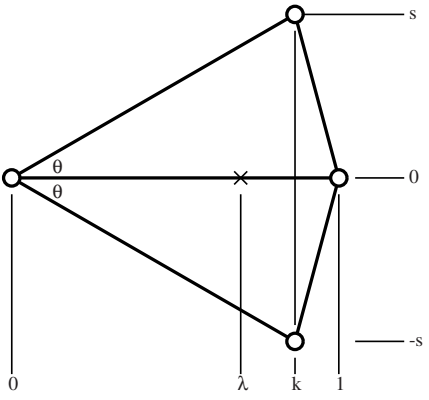


Fig. 8. Plan view of natural configuration

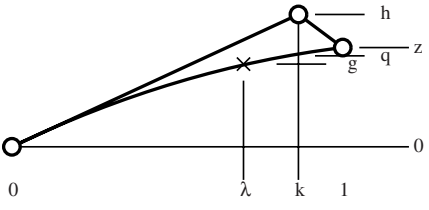


Fig. 9. Side view of natural configuration

6.1 Loop

e-vertices. Consider two adjacent sectors of the inner part of the natural configuration, as shown in Figure 8.

The extraordinary vertex, of valency n, is at the left, and the others are at unit distance from it in the abscissa plane. λ is here used to denote the shrinkage factor from one step to the next. This is, of course, equal to the $\omega = 1$ eigenvalue, but we are not using that definition explicitly here.

$$\begin{aligned} \theta &= 2\pi/n, \\ s &= \sin(\theta), \\ k &= \cos(\theta). \end{aligned} \tag{9}$$

Without loss of generality, we can subtract any linear function from the ordinates of the control net. Let the ordinate at the extraordinary vertex (at the left of the above figure) be zero, that at the far end of the central edge be z and those at the two 'wing' vertices be h. (These two ordinates can be made equal by subtraction of a function linear in y.) See Figure 9; the curved line is a cross-section through a quadratic surface of zero mean curvature.

Let the ordinate at the place where the line joining the wings crosses the edge be q, and that at λ be g. $[\lambda, 0, g]$ will be the place where we insert a new e-vertex.

We are looking for a formula giving g as a weighted mean of 0, h, and z and for implementation reasons we would like the ratio of the coefficients of h and z to remain in the same ratio as in the regular case.

$$\begin{aligned}
\delta_{xx} &= \left(\frac{z-q}{1-k} - \frac{q}{k}\right)/1, \\
&= \frac{kz - kq - q + qk}{k(1-k)}, \\
&= \frac{kz - q}{k(1-k)}, \\
\delta_{yy} &= \left(\frac{h-q}{s} - \frac{q-h}{s}\right)/2s, \\
&= \frac{h-q}{s^2}, \\
&= \frac{h-q}{1-k^2}, \\
&= \frac{h-q}{(1-k)(1+k)}, \\
2m &= \delta_{xx} + \delta_{yy}, \\
&= \frac{kz-q}{k(1-k)} + \frac{h-q}{(1-k)(1+k)}, \\
&= \left(\frac{(kz-q)(1+k) + k(h-q)}{k(1-k)(1+k)}\right).
\end{aligned} \tag{10}$$

This will be zero when

$$\begin{aligned}
0 &= (kz-q)(1+k) + k(h-q), \\
&= k(1+k)z - (1+k)q + kh - kq, \\
&= k(1+k)z + kh - kq - (1+k)q, \\
&= k(1+k)z + kh - (1+2k)q, \\
q &= \frac{k(1+k)}{1+2k}z + \frac{k}{1+2k}h.
\end{aligned} \tag{11}$$

We can then determine the value g to be inserted at λ from:

$$\begin{aligned}
\frac{\lambda z - g}{\lambda(1-\lambda)} &= \frac{kz-q}{k(1-k)} \,(= \delta_{xx}), \\
\frac{g}{\lambda(1-\lambda)} &= \frac{z}{1-\lambda} - \frac{kz-q}{k(1-k)}, \\
&= z\left(\frac{1}{1-\lambda} - \frac{k}{(1-k)(1+2k)}\right) + \frac{h}{(1-k)(1+2k)}.
\end{aligned} \tag{12}$$

It appears unlikely, but this complicated expression is a self-fulfilling prophecy for λ. If we set $z = 1$ and $h = k$, so that the far vertex and the two wings are coplanar with the EV, the value of q also pans out as h, and g appears as λ. We cannot therefore use the condition $g = \lambda$ to determine λ. Thus we can choose whatever value of λ we like. We will always find $\mu_2 = \lambda^2$.

We therefore need some other criterion for choosing λ. If we choose $\lambda = 1/2$, the triangles near the EV will shrink at the same rate as others. However, for implementation it is significantly better to use the value of λ which lets the contributions of z and h to g to remain in the same ratio as for the regular case. This means that the relative contributions of the h- and z-points do not depend on the valency of any other vertex, and we can set up a system in which the contribution weights of the four relevant vertices depend only on their own valencies. These are then normalised to give a weighted mean. This means that close EVs (and extraordinary faces) are all handled by a single, simple piece of code.

The relative weights of z and h are $\left(\frac{1}{1-\lambda} - \frac{k}{(1-k)(1+2k)}\right)$ and $\frac{1}{(1-k)(1+2k)}$. We want the ratio of these to be 3:2 to match the effective ratio in the regular case.

$$\begin{aligned}
\left(\frac{1}{1-\lambda} - \frac{k}{(1-k)(1+2k)}\right) &= \frac{3}{2(1-k)(1+2k)}, \\
\frac{2}{1-\lambda} &= \frac{2k}{(1-k)(1+2k)} + \frac{3}{(1-k)(1+2k)}, \\
&= \frac{2k+3}{(1-k)(1+2k)} \\
1-\lambda &= \frac{2(1-k)(1+2k)}{3+2k}, \\
\lambda &= \frac{1+4k^2}{3+2k}.
\end{aligned} \tag{13}$$

This, not surprisingly, turns out to be equal to the value $\frac{3+2\cos(4\pi/n)}{3+2\cos(2\pi/n)}$ which is derived in the Appendix straightforwardly from the Fourier approach. Substituting this back into the generic stencil for the mean second divided difference gives the weights for the new e-vertices on the rays from extraordinary vertices.

If we take the weights from the neighbouring ordinary vertices to be 6 and 2 respectively, and the weight from the extraordinary one to be β (using a normalised weighted mean dividing all by $(10 + \beta)$), we require that

$$\frac{\lambda(1-\lambda)}{(1-k)(1+2k)} = \frac{4}{10+\beta} \tag{14}$$

$$\begin{aligned} 10+\beta &= \frac{4(1-k)(1+2k)}{\lambda(1-\lambda)}, \\ \text{but } \lambda &= \frac{1+4k^2}{3+2k} \\ \text{and } 1-\lambda &= \frac{2(1-k)(1+2k)}{3+2k}, \\ \text{and so } 10+\beta &= \frac{(1-k)(1+2k)(3+2k)^2}{(1+4k^2)(1-k)(1+2k)} \\ &= \frac{2(3+2k)^2}{1+4k^2} \\ \beta &= \frac{8(1-k)(1+4k)}{1+4k^2}. \end{aligned} \tag{15}$$

Thus the computation of the e-vertices is just a weighted mean where the weight associated with a given old vertex is an appropriate function $\beta(n)$ of its valency.

The reason why Loop in its original form fails the $\mu_2 = \lambda^2$ criterion and thus does not have bounded curvature, is that it uses the regular stencil for the e-vertices instead of this slightly more complicated one.

v-vertices. We then set the v-vertices by equating the mean second difference at each new v-vertex to the mean second difference on the old mesh at its ancestor.

Writing A and $\bar{B}$ for the extraordinary vertex and the mean of the 1-ring vertices before refinement and a and $\bar{b}$ for the corresponding points after, we have the equation

$$a - \bar{b} = \lambda^2(A - \bar{B}).$$

This may be considered the easiest form for implementation. Measure the mean second divided difference at each old vertex before refinement, scale it by λ^2 (which depends on the valency) and then after the e-vertices are determined, add this displacement on to the mean of the new 1-ring to get the new v-vertex.

However, because we know that $\bar{b} = \frac{10\bar{B}+\beta A}{10+\beta}$, we can either substitute for $\bar{b}$ in the equation for a to show that the convex hull property is maintained, or else substitute for $\bar{B}$ to give a as a combination of A and $\bar{b}$. This formulation avoids the need to allocate memory for holding the mean second differences. In this case, by associating the standard weight of 1 with each 1-ring vertex, the construction of a is again a normalised weighted mean where the weights depend on the valencies of the points they are associated with. The weight, α, for A becomes $\lambda^2(10+\beta)/10$. The values of $\alpha(n)$ and $\beta(n)$ are those that result from eigenanalysis-based design. They are tabulated in Table 2.

For valencies 5 or greater, this is all that is necessary to achieve bounded second derivatives. However, the same principle can be applied at the new e-vertices on the opposite edge of a triangle to an extraordinary vertex. See Figure 10.

The same process is applied to determine the relative weights of the e-vertices on the diagonal of the kite, but this time we already know λ from the previous computations. But we do have a variable available for choosing, which is the

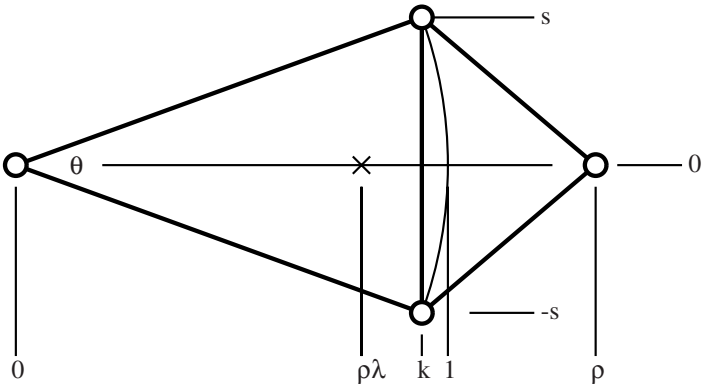


Fig. 10. The kite formed from two triangular faces with an extraordinary vertex at the left

ratio ρ of the distances from the 2-ring and the 1-ring to the EV. This can be set so that the new e-vertex also satisfies the zero mixed mean second divided difference criterion.

For valencies 3 and 4 this definitely needs to be done, and for valency 6 the regular case gives the same answer. It does not need to be done for other valencies from the point of view of bounded curvature (this e-vertex will no longer be part of the ring in just two more steps) but the argument of making a well-shaped natural configuration suggests that it could well be applied for all valencies. It has the effect of determining the appropriate weight of the EV in the linear combination giving the new far e-vertex.

It is also possible to apply the same logic to the new 1-ring v-vertices.

Thus this approach can determine all of the entries in the EV mask, without resorting to optimisation. It remains to be seen whether the surfaces so generated are as good as the optimised ones.

6.2 Catmull-Clark

f-vertices and e-vertices. In the Catmull-Clark case, we have two freedoms, ρ and λ, as shown in Figure 11. As well as demanding that the relative contributions of the regular vertices should remain in the standard ratio, we can also ask that $\rho^2 = 1/\lambda$. This gives the property that in the natural configuration the diagonal quads formed from the original vertices and the f-vertices should be the same shape as the quads before and after a refinement step. Again, this seems a plausible heuristic with keeping the well-formedness of the natural configuration in mind. It also leads to the programming convenience of being able to use the same weight for the EV in both the f-vertex and e-vertex steps.

This is the heuristic used by Sabin [13]. Note that it is not the same as using the same weight in the mask for both e- and f-vertices. Indeed, this multistage construction is not quite representable as mask tuning, because the influence of

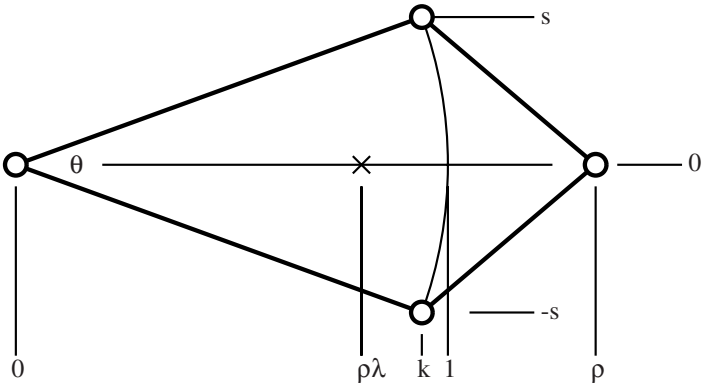


Fig. 11. The kite formed from one quad face

the 1-ring on e-vertices comes via two paths, one of which is already influenced by the EV weight. This probably explains why the optimum λ values found by Augsdörfer [1] are not quite the same as the values found here. See Figure 11.

Applying similar arguments to those used for the Loop case above, we find that

$$\begin{aligned} q\rho(k\rho - k^2 + s^2) &= k\rho(\rho - k)h + ks^2 z, \\ \frac{g}{\rho\lambda(1-\lambda)} &= \left(\frac{1}{\rho(1-\lambda)} - \left(\frac{k}{k\rho - k^2 + s^2}\right)\right) z + \left(\frac{\rho}{k\rho - k^2 + s^2}\right) h. \end{aligned} \tag{16}$$

In this case the coefficient of h should be twice that of z, and substituting $s^2 + k^2 = 1$ and $\lambda = 1/\rho^2$, we find that

$$\rho^3 + \rho(4k^2 - 3) - 2k = 0.$$

If we assume that 4-valent vertices have unit weight in the normalised weighted means for the f-vertices and v-vertices, we find that the weight $W(n)$ for vertices of valency n should be $\rho(\rho + 2k) - 3$. The same cubic equation and the same weight were derived by Sabin [13] by solving $\mu_2 = \lambda^2$.

v-vertices. We can again use the equation

$$m_g + 2m_a = M_g + 2M_a, \tag{17}$$

where

$$\begin{aligned} M_g &= \lambda^2(\bar{B} - A), \\ M_a &= \lambda^2 \left(\frac{\bar{C} - A}{\rho^2}\right), \\ m_g &= (\bar{b} - a), \\ m_a &= \left(\frac{\bar{c} - a}{\rho^2}\right). \end{aligned} \tag{18}$$

Table 1. Coefficients for Catmull-Clark at Extraordinary Vertices

n	$W(n)$	$\alpha(n)$	$\beta(n)$	$\gamma(n)$
3	1.236068	0.170033	0.751865	0.078102
4	0.999999	0.25	1	−0.25
5	0.718502	0.344048	1.340796	−0.684845
6	0.522333	0.440062	1.677230	−1.117293
7	0.391842	0.527227	1.966112	−1.493339
8	0.303196	0.601358	2.200535	−1.801893
9	0.240967	0.662508	2.387118	−2.049626
10	0.195860	0.712380	2.535252	−2.247632
11	0.162219	0.753001	2.653472	−2.406473
12	0.136502	0.786215	2.748627	−2.534842

This leads to the equation for a becoming

$$\begin{aligned} \rho^2\bar{b} + 2\bar{c} - (\rho^2+2)a &= \lambda^2(\rho^2\bar{B} + 2\bar{C} - (\rho^2+2)A), \\ (\rho^2+2)a &= \rho^2\bar{b} + 2\bar{c} + \lambda^2(\rho^2+2)A - \lambda^2\rho^2\bar{B} - 2\lambda^2\bar{C}. \end{aligned} \tag{19}$$

Again, we can either substitute for $\bar{b}$ and $\bar{c}$ in terms of $\bar{B}$ and $\bar{C}$ to demonstrate the convex hull property, or else substitute for $\bar{B}$ and $\bar{C}$ to give an expression $a = \alpha A + \beta\bar{b} + \gamma\bar{c}$ in terms of the new local points only.

In fact, for simplicity, we could ignore the ρ^2 factor, leaving the value at 2 irrespective of the valency of A. This also gives a convex combination for all valencies.

Strictly we should address the regularity of the characteristic map, and also prove that the difference between the two estimators converges to zero as the refinement proceeds. These proofs are omitted for brevity.

Note that the negative values of γ do not indicate a failure of the convex hull property. a in terms of A, $\bar{B}$ and $\bar{C}$ is a positive linear combination.

6.3 4-8

f-vertices. The f-vertex step of the 4-8 scheme now becomes identical with that of Catmull-Clark. The argument about having the shape of the quad the same at the $\sqrt{2}$ stage is even more plausible here, of course.

This leads to the slightly surprising results that the natural configuration is the same as for the Catmull-Clark variant described above, and that the shrinkage factor (λ) for the two steps taken together to give a binary scheme rather than $\sqrt{2}$ is also the same.

v-vertices. We cannot just use the simplistic rule of still equating the new St. Georges estimator to the old St. Andrews one. This runs immediately into coefficients which fail the convex hull test. The equivalent influence of $\bar{C}$ is zero for the regular case, and it becomes negative for $n > 4$. Instead we equate the new St.Georges estimator to a linear combination of the two old estimators such that the equivalent influence of $\bar{C}$ is zero.

7 Conclusions and Loose Ends

By looking at the construction of the coefficients of subdivision schemes near the extraordinary points in terms of second divided differences, we are able to achieve bounded curvature using geometric arguments in place of Fourier components and eigenanalysis. In each case this is achieved in a multi-stage algorithm where the weight of a old vertex in each weighted mean depends only on its valency, thus leading to a very clean implementation, in which valency impacts only through the values of weights looked up in a set of simple tables. There is no special case code for extraordinary vertices.

The variants so created here are not particularly new, but this approach leads in a number of promising directions, not yet explored.

- Better tuning can be done by using second divided differences at new vertices outside the 1-ring. We think it reasonable to expect that if using weights dependent on the valency of the extraordinary vertex can give bounded curvature so easily, using the same principles further out may give a reduction in the variation of curvature over the spline rings. In order to achieve C2 from a generalisation of the bicubic B-spline it would be necessary to alter the rules at all vertices, not just the local ones, but going a little further may well make an improvement to the local shape.
- In fact it is possible to go further than that, by observing that in the various constructions we have shown here, the valency is not directly involved. What has a direct effect is the angle of a face at that vertex. We conjecture that by using the actual angles (rather than $2\pi/n$) we could even achieve C2. This would, of course, be a non-stationary geometry sensitive scheme.
- It is also possible to replace second divided differences by estimators of curvature, in the style of [14], which we hope will lead us to non-stationary schemes which preserve natural quadrics. That, however, is another story.

References

1. Augsdorfer, U., Dodgson, N.A., Sabin, M.A.: Tuning Subdivision by Minimising Gaussian Curvature Variation Near Extraordinary Vertices. Eurographics 2006, pp. 263–272 (2006)
2. Barthe, L., Gerot, C., Sabin, M.A., Kobbelt, L.: Eigencomponents of a Subdivision Matrix in the Fourier Domain. In: Adv. Multiresolution for Geometric Modelling, pp. 245–258. Springer, Heidelberg (2005)
3. Catmull, E., Clark, J.: Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. CAD 10(6), 183–188 (1978)
4. Doo, D., Sabin, M.: Behaviour of Recursive Subdivision Surfaces Near Extraordinary Points. CAD 10(6), 189–194 (1978)
5. Dyn, N.: Analysis of Convergence and Smoothness by the Formalism of Laurent Polynomials. In: Tutorials on Multiresolution in Geometric Modelling, pp. 51–68. Springer, Heidelberg (2002)
6. Dyn, N., Levin, D., Gregory, J.A.: A Four Point Interpolatory Subdivision Scheme for Curve Design. Computer Aided Geometric Design 4(4), 257–268 (1987)

7. Dyn, N., Levin, D., Gregory, J.A.: A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. ACM Trans. Graphics 9(2), 160–169 (1990)
8. Eastlick, M.: PhD Dissertation. Sheffield University (2006)
9. Kobbelt, L.: Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. Computer Graphics Forum 15, 409–420 (1996)
10. Loop, C.: Smooth Subdivision Surfaces Based on Triangles. Master's Thesis, Department of Mathematics, University of Utah (1987)
11. Peters, J., Reif, U.: Shape Characteristics of Subdivision Surfaces: Basic Principles. Computer Aided Geometric Design 21, 515–599 (2004)
12. Reif, U.: A Degree Estimate for Subdivision Surfaces of Higher Regularity. Proc. AMS 124(7), 2167–2174 (1996)
13. Sabin, M.: Cubic Recursive Subdivision with Bounded Curvature. In: Schumaker, Laurent, LeMehaute (eds.) Curves and Surfaces, pp. 415–420. Academic Press, San Diego (1991)
14. Sabin, M., Dodgson, N.: A Circle-preserving Variant of the Four-point Subdivision Scheme. In: Mathematical Methods for Curves and Surfaces, Daehlen, Mörken, Schumaker (eds.) Nashboro Press, pp. 275–286 (2004)
15. Velho, L., Zorin, D.: 4-8 Subdivision. Computer Aided Geometric Design 18, 397–427 (2001)

Appendix: Tuning Loop to Bounded Second Derivative by Eigenanalysis of Fourier Partitions

Let the configuration around the extraordinary vertex A before refinement be indicated by Figure 12(a), the configuration after refinement by Figure 12(b), and the mask by which A influences the nearby new vertices be as indicated in Figure 12(c). General valency is indicated by the incompleteness of these figures.

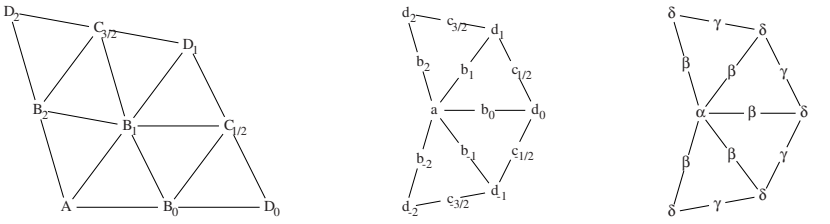
We assume that all other local vertices are regular, because we are concerned with the limit, when the extraordinary vertices will have become separated. They therefore contribute to the new vertices using the normal mask, shown in Figure 4.

The equations for a partition ω are set up by assuming that all the original vertices are of the form $B_j = B\cos(2\pi j\omega/n)$ where n is the valency. The new valencies are then found by symmetry to have the same form, and the different Fourier components are thus separated.

The equations for the $\omega \neq 0$ partitions are

$$\begin{bmatrix}(\beta+10)b\\(\gamma+14)c\\(\delta+15)d\end{bmatrix} = \begin{bmatrix}6+4\cos(2\pi\omega/n) & 0 & 0\\ 12\cos(\pi\omega/n) & 2 & 0\\ 10+2\cos(2\pi\omega/n) & 2\cos(\pi\omega/n) & 1\end{bmatrix}\begin{bmatrix}B\\C\\D\end{bmatrix}. \tag{20}$$

and we can read the eigenvalues off by taking the diagonal elements and dividing by the normalisation factors (e.g. $\beta + 10$) on the left hand side. The dominant eigenvalue is from the top row, (which can be confirmed) and so we find that



(a) (b) (c)

Fig. 12. Configuration around an extraordinary vertex

Table 2. Coefficients for Bounded Curvature Loop

n	α	β	λ
4	5	8	0.333333
5	8.090169	8.944271	0.381966
6	10	6	0.5
7	13.211023	4.119085	0.601594
8	17.980375	2.990187	0.679622
9	24.415671	2.272489	0.738576
10	32.652475	1.788854	0.783457
11	42.849172	1.447060	0.818115
12	55.176914	1.196152	0.845299

$$\begin{aligned} \lambda &= \frac{6 + 4\cos(2\pi/n)}{\beta + 10}, \\ \mu_2 &= \frac{6 + 4\cos(4\pi/n)}{\beta + 10}. \end{aligned} \tag{21}$$

Writing k_ω for $\cos(\pi\omega/n)$, and equating $\mu_2 = \lambda^2$ gives

$$\begin{aligned} \beta + 10 &= \frac{(6 + 4k_2)^2}{6 + 4k_4}, \\ \lambda &= \frac{6 + 4k_4}{6 + 4k_2}. \end{aligned} \tag{22}$$

The equation for the $\omega = 0$ partition is

$$\begin{bmatrix} (\alpha + n)a \\ (\beta + 10)b \\ (\gamma + 14)c \\ (\delta + 15)d \end{bmatrix} = \begin{bmatrix} \alpha & n & 0 & 0 \\ \beta & 10 & 0 & 0 \\ \gamma & 12 & 2 & 0 \\ \delta & 12 & 2 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}. \tag{23}$$

The top 2x2 block has eigenvalues 1 and $\mu_0 = \frac{10\alpha - n\beta}{(\alpha+n)(\beta+10)} = \lambda^2$, a linear equation in α, which may be solved using the values of β and λ just determined to give

$$\alpha = \frac{n(\beta + \lambda^2(\beta + 10))}{10 - \lambda^2(\beta + 10)}.$$

The values of γ and δ are not constrained by this approach, except in the special cases of $n = 4$, when γ has to give a second $\mu_2 = \lambda^2$ condition, and of $n = 3$, when the saddle eigencomponent comes from $\omega = \pm 1$ and has to be constrained using γ. δ is not constrained, and it can reasonably be set to $\gamma/2$.

However, to get a well-formed natural configuration and minimal variation of curvature tuning of γ and δ is desirable.

Facial Shape-from-Shading Using Principal Geodesic Analysis and Robust Statistics

William A.P. Smith and Edwin R. Hancock

Department of Computer Science, The University of York, UK
{wsmith,erh}@cs.york.ac.uk

Abstract. In this paper we make two contributions to the problem of recovering surface shape from single images of faces. The first of these is to develop a representation of the distribution of surface normals based on the exponential map, and to show how to model shape-deformations using principal geodesic analysis on the exponential map. The second contribution is to show how ideas from robust statistics can be used to fit the model to facial images in which there is significant self-shadowing. The method is evaluated on both synthetic and real-world images. It is demonstrated to effectively fill-in the facial surface when more than 30% of the area is subject to self-shadowing.

1 Introduction

The problem of reconstructing the surface height function of a face from a single image is a challenging one that has eluded efforts in shape-from-shading for several decades [1,2,3,4,5,6]. The goal is to use the image irradiance equation to recover estimates of local surface orientation, and then to recover surface height by integrating the field of surface normals. Unfortunately, there are a number of well documented problems that frustrate this task [1]. The first of these is that the recovered field of surface normals is subject to concave-convex shape ambiguities, and this can lead to the implosion of facial features. This effect can lead to facial features becoming inverted (such as the nose), and the exaggeration of the relief of others (for instance the cheeks). The second problem is that of self shadowing. This occurs when the light source is at an oblique angle to the face, and the nose casts a shadow and the eye-sockets are also in shadow.

It is for these reasons that the problem of facial shape-from-shading is addressed either using domain specific constraints or using a face-specific shape-model. For instance Zhao and Chellappa [1] have shown how improved shape recovery can be achieved using a symmetric shape-from-shading method. Atick et al [7] have used a model based method constructed from range-data. Smith and Hancock [8] have developed a statistical model of surface orientation distribution trained on gradient data from range images. The model is fitted to intensity data using geometric constraints on the surface normal direction provided by Lambert's law. In related work, Vetter and Blanz [6] have shown how recognition can be performed by fitting a surface model to brightness data, and

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 412–426, 2007.
© Springer-Verlag Berlin Heidelberg 2007

how the fitted model can be used for realistic view synthesis [9]. Finally, it is worth noting that if multiple images from a fixed viewing direction and variable light source direction are available, then photometric stereo can be used for accurate facial surface recovery [2].

Although the methods listed above lead to realistic shape recovery, they do not work well when there is significant self shadowing. Our aim in this paper is therefore to develop a statistical model that can be fitted to facial images where there is significant degree of self-shadowing. The model is trained on surface normal data acquired from sets of range images. The images are aligned, and the model is couched in terms of the distribution of surface normal directions at each aligned image location. Our representation of the distribution of normal directions is based on the exponential map from a unit sphere onto a Cartesian plane. To capture the statistics of the distribution of the surface normals under the exponential map, we borrow ideas from principal geodesic analysis. This is a technqiue first proposed by Fletcher, Joshi, Lu and Pizer [10] for the analysis of tensor MRI data, and recently extended to the genernal problem of statistical analysis on Riemannian manifolds by Pennec [11].

With the model to hand, we demonstrate how it may be fitted to image brightness data, so as to satisfy constraints provided by Lambert's law. The resulting iterative update rule for the fitted surface normal is particularly simple and can be couched in terms of rotation operations on the exponential map. To deal with self shadowing, we extend the fitting method to deal with outliers using the apparatus of robust statistics. We use the difference between measured and fitted brightness values to compute a shadow-weight. The weights are used to exclude shadow regions in the shape-parameter estimation process. The parameter update scheme is based on M-estimators.

We experiment with the resulting shape-recovery method on both synthetic images with known ground truth and real-world images from the Yale-B database. The results indicate that the method works well even when the angle between the light source and the viewing direction exceeds 60°. Moreover, the fitting method is able to reliably fill-in shadowed regions of the face.

2 Shape-from-Shading

Let I_p be the normalised image brightness at pixel p, $\mathbf{n}_p$ be the local surface normal and $\mathbf{s}$ be a vector describing the direction of a single, distant light source. According to Lambert's law $I_p = \mathbf{n}_p.\mathbf{s}$. In general, the surface normal $\mathbf{n}_p$ can not be recovered from a single brightness measurement since it has two degrees of freedom corresponding to the elevation and azimuth angles on the unit sphere. However, compliance with Lambert's law can be ensured by constraining the recovered surface normal to lie on the reflectance cone whose axis is aligned with the light-source vector $\mathbf{s}$ and whose opening angle is $\theta_p = \arccos I_p$. At each iteration the surface normal is free to move to an off-cone position subject to smoothness or curvature consistency constraints. However, the hard irradiance constraint is re-imposed by rotating each surface normal back to its closest

on-cone position. This process ensures that the recovered field of surface normals satisfies the image irradiance equation after every iteration.

Suppose that $\mathbf{n}_p'^{(t)}$ is an off-cone surface normal at iteration t of the algorithm, then the update equation is $\mathbf{n}_p^{(t+1)} = \Theta_p \mathbf{n}_p'^{(t)}$, where Θ_p is a rotation matrix computed from the apex angle α_p and the angle between $\mathbf{n}_p'^{(t)}$ and the light source direction $\mathbf{s}$. To restore the surface normal to the closest on-cone position it must be rotated by an angle $\alpha_p = \theta_p - \arccos\left[\mathbf{n}_p'^{(t)}.\mathbf{s}\right]$ about the axis $(u, v, w)^T = \mathbf{n}_p'^{(t)} \times \mathbf{s}$. Hence, the rotation matrix is

$$\Theta_p = \begin{pmatrix} c + u^2c' & -ws + uvc' & vs + uwc' \\ ws + uvc' & c + v^2c' & -us + vwc' \\ -vs + uwc' & us + vwc' & c + w^2c' \end{pmatrix}, \tag{1}$$

where $c = \cos(\alpha_p)$, $c' = 1 - c$ and $s = \sin(\alpha_p)$. This method of imposing Lambert's law requires that each surface normal is initialised to a position on their reflectance cones. In the absence of any model-based constraint with which to choose the initialisation, this may be done by placing the surface normals on their reflectance cones such that they are aligned in the direction opposite to that of the local image gradient. This initialisation is consistent with the assumption that the object under study is globally convex. The polar angle $\phi_p = \arctan\{(-\partial I/\partial y)/(-\partial I/\partial x)\}$ of the local negative image gradient at the point p can be used to initialise the surface normal directions by setting $\mathbf{n}_p^{(0)} = (\sin\theta_p \cos\phi_p, \sin\theta_p \sin\phi_p, \cos\theta_p)^T$.

3 A Statistical Surface Normal Model

We aim to augment shape-from-shading with a statistical model which compactly captures variations in local surface orientation across a face. Most statistical shape models rely on Principal Components Analysis (PCA) to construct an efficient parameterisation of shape variation [7]. However, since a linear combination of unit vectors is not a unit vector, the analysis of the distribution of surface normals thus cannot be effected in a linear way. We therefore choose to model a distribution of surface normals as a distribution of points lying on a spherical manifold and turn to the Principal Geodesic Analysis (PGA) approach of Fletcher et al. [10] in order to build our statistical model. In this section we describe PGA, demonstate how it may be applied to a set of needle-maps and then show how we use the principal geodesics to deform the field of surface normals.

3.1 Preliminaries

Spherical Medians: A distribution of spherical directional data $\mathbf{n}_1, \ldots, \mathbf{n}_N \in \Re^3$ can be characterised using the mean direction [12]: $\overline{\mathbf{n}}_0 = \overline{\mathbf{n}}/\|\overline{\mathbf{n}}\|$ where $\overline{\mathbf{n}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{n}_i$. This is the Euclidian mean (or centre of mass) of the distribution of

points, projected back onto the closest point on the sphere. In this formulation, each unit vector $\mathbf{n}_i$ is treated as a point in Euclidian space. The mean direction is therefore the *extrinsic mean* of the distribution, since it is based on minimising the Euclidian distance to each point:

$$\overline{\mathbf{n}} = \arg\min_{\mathbf{n}\in\Re^3} \sum_{i=1}^{N} \|\mathbf{n} - \mathbf{n}_i\|. \tag{2}$$

A more natural representation is to treat each datum as a point on a unit sphere: $\mathbf{n}_1, \ldots, \mathbf{n}_N \in S^2$ and use arc length as the choice of distance measure. Since a 2-sphere is a Riemannian manifold and great circles are geodesics, this is the Riemannian distance $d(.,.)$ between a pair of points, $d(\mathbf{n}_1, \mathbf{n}_2) = \arccos(\mathbf{n}_1.\mathbf{n}_2)$. Using this definition of distance, we can define the *intrinsic mean*:

$$\boldsymbol{\mu} = \arg\min_{\mathbf{n}\in S^2} \sum_{i=1}^{N} d(\mathbf{n}, \mathbf{n}_i). \tag{3}$$

For a spherical manifold, this point is known as the spherical median [13] and can be found using the gradient descent method of Pennec [11].

The Log and Exponential Map: If $v \in T_{\mathbf{n}}S^2$ is a vector on the tangent plane to S^2 at $\mathbf{n}$ and $v \neq 0$, the *exponential map*, denoted $\mathrm{Exp}_{\mathbf{n}}$, of v is the point on S^2 along the geodesic in the direction of v at distance $\|v\|$ from $\mathbf{n}$. Geometrically, this is equivalent to marking out a length equal to $\|v\|$ along the geodesic that passes through $\mathbf{n}$ in the direction of v. The point on S^2 thus obtained is denoted $\mathrm{Exp}_{\mathbf{n}}(v)$. This is illustrated in Fig. 1. The inverse of the exponential map is the

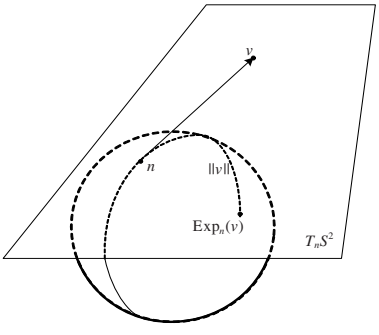


Fig. 1. The exponential map

log map, denoted $\mathrm{Log}_{\mathbf{n}}$. The geodesic distance between two points $\mathbf{n}_1$ and $\mathbf{n}_2$ can be expressed in terms of the log map, i.e. $d(\mathbf{n}_1, \mathbf{n}_2) = \|\mathrm{Log}_{\mathbf{n}_1}(\mathbf{n}_2)\|$.

Variance on a Sphere: We can use the log map and intrinsic mean to define the sample variance of a distribution of points on the sphere:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{\mu}, \mathbf{n}_i)^2 = \frac{1}{N}\sum_{i=1}^{N} \|\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{n}_i)\|^2.$$

Projection onto Principal Geodesics: In standard PCA, the lower-dimensional subspaces form a linear subspace of the space in which the data lies. In PGA, this notion is replaced by a *geodesic submanifold*. In other words, whereas each principal axis in PCA is a straight line, in PGA each principal axis is a geodesic curve. In the spherical case this corresponds to a great circle.

In PCA a datum may be projected onto a subspace using linear projection. In the case of PGA, a different kind of projection operator is required. Consider a great circle G on the sphere S^2. In order to project a point $\mathbf{n}_1 \in S^2$ onto the point on G that is closest to $\mathbf{n}_1$ in terms of Riemannian distance, we use the projection operator $\pi_G : S^2 \rightarrow G$ given by $\pi_G(\mathbf{n}_1) = \arg\min_{\mathbf{n}\in G} d(\mathbf{n}_1, \mathbf{n})^2$. For a geodesic G passing through the spherical median $\boldsymbol{\mu}$, we may approximate π linearly in the tangent plane to S^2 at $\boldsymbol{\mu}$, i.e. $T_{\boldsymbol{\mu}}S^2$. If $v_1, ..., v_K$ is an orthonormal basis for $T_{\boldsymbol{\mu}}S^2$, then the projection operator π can be approximated in the tangent plane using:

$$\mathrm{Log}_{\boldsymbol{\mu}}\left(\pi_G(\mathbf{n}_1)\right) \approx \sum_{i=1}^{K} v_i.\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{n}_1). \tag{4}$$

Computing Principal Geodesics: With the preliminaries above to hand, the principal geodesics of $\mathbf{n}_1, \ldots, \mathbf{n}_N \in S^2$ may be computed. By analogy with PCA, the goal of PGA is to find a sequence of nested geodesic submanifolds which account for descreasing variance of the data. The principal geodesics are defined by first constructing an orthonormal basis of tangent vectors $v_1, .., v_K \in T_{\boldsymbol{\mu}}S^2$. These are the principal directions on the tangent plane $T_{\boldsymbol{\mu}}S^2$. The ith principal geodesic is given by $\mathrm{Exp}_{\boldsymbol{\mu}}(v_i)$.

Using the approximation (4), the first principal direction is given by:

$$v_1 \approx \arg\max_{\|v\|=1} \sum_{i=1}^{N} \left(v.\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{n}_i)\right)^2. \tag{5}$$

the remaining principal directions are defined recursively as:

$$v_k \approx \arg\max_{\|v\|=1} \sum_{i=1}^{N}\sum_{j=1}^{k-1} \left(v_j.\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{n}_i)\right)^2 + \left(v.\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{n}_i)\right)^2. \tag{6}$$

This minimisation problem is equivalent to the standard principal components analysis of the vectors $\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{n}_i)$ in $T_{\boldsymbol{\mu}}S^2$. In the following section we show how to apply principal geodesic analysis to a set of exemplar facial needle-maps for the purposes of learning a statistical model of face-shape.

3.2 PGA of Needle-Maps

Suppose that each of the K training examples is a range image which consists of an array of depth data of dimension N. For the pixel indexed p in the kth training sample the depth is z_p^k. Using the range data we estimate the surface normal directions, and the surface normal at the pixel location p for the kth training image is $\mathbf{n}_p^k = \left((\mathbf{n}_p^k)_x, (\mathbf{n}_p^k)_y, (\mathbf{n}_p^k)_z\right)^T$, where $(\dots)_x$ denotes the x component of the parenthesised vector, and the y and z components are defined similarly.

We calculate the spherical median $\boldsymbol{\mu}_p$ of the distribution of surface normals $\mathbf{n}_p^1, \dots, \mathbf{n}_p^K$ at each pixel location p. The surface normal $\mathbf{n}_p^k$ is represented by its position on the tangent plane at $\boldsymbol{\mu}_p$ given by the log map: $\mathbf{v}_p^k = \mathrm{Log}_{\boldsymbol{\mu}_p}(\mathbf{n}_p^k)$.

The covariance matrix $\mathbf{L}$ for the data is given by:

$$\mathbf{L} = \frac{1}{K}\mathbf{D}\mathbf{D}^T, \tag{7}$$

where $\mathbf{D} = [\mathbf{v}^1|\dots|\mathbf{v}^K]$ and $\mathbf{v}^k = [\mathbf{v}_1^k, \dots, \mathbf{v}_N^k]^T$. We use the numerically efficient *snap-shot* method of Sirovich to compute the eigenvectors of $\mathbf{L}$. Accordingly, we construct the matrix $\hat{\mathbf{L}} = \frac{1}{K}\mathbf{D}^T\mathbf{D}$ and find the eigenvalues and eigenvectors. The ith eigenvector $\hat{\mathbf{e}}_i$ of $\hat{\mathbf{L}}$ can be used to find the ith eigenvector $\mathbf{e}_i$ of $\mathbf{L}$ using $\mathbf{e}_i = \mathbf{D}\hat{\mathbf{e}}_i$. The eigenvalues of $\mathbf{L}$ may also be found using the eigen-decomposition of $\hat{\mathbf{L}}$. Let λ_i be the ith eigenvalue of $\mathbf{L}$ and $\hat{\lambda}_i$ the ith eigenvalue of $\hat{\mathbf{L}}$. For $i \leq K$, $\lambda_i = \hat{\lambda}_i$. For $i > K$, $\lambda_i = 0$.

Provided the effects of noise are small, we need only retain S eigenmodes to retain p percent of the model variance. We choose S to be the smallest integer that satisfies the condition: $\sum_{i=1}^{S} \lambda_i \geq \frac{p}{100}\sum_{i=1}^{K} \lambda_i$.

Given a vector of parameters $\mathbf{b} = [b_1|b_2|\dots|b_S]$ we can generate a needle-map using:

$$\mathbf{n}_p = \mathrm{Exp}_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{b})_p\right). \tag{8}$$

We use the notation $(\dots)_p$ to denote the selection of the pth tangent vector from the parenthesised column vector, e.g. $\left([\mathbf{v}_1, \dots, \mathbf{v}_N]^T\right)_p = \mathbf{v}_p$.

4 Incorporating Principal Geodesics into SFS

In this section we show how our statistical surface normal model may be incorporated into an iterative shape-from-shading algorithm. We use the model to update the field of surface normals by projecting the estimated normals onto the span of the principal geodesics. We subsequently ensure data-closeness by rotating each normal back onto its reflectance cone and iterate the process. If $\mathbf{n}_p^{(t)}$ is the estimated surface normal at pixel p at iteration t, we write $\mathbf{v}_p^{(t)} = \mathrm{Log}_{\boldsymbol{\mu}_p}(\mathbf{n}_p^{(t)})$. The vector of parameters $\mathbf{b}^{(t)}$ which describes the best-fit in the model space to the field of surface normals at iteration t is given by

$$\mathbf{b}^{(t)} = \mathbf{P}^T\mathbf{v}^{(t)}, \tag{9}$$

where $\mathbf{v}^{(t)} = [\mathbf{v}_1^{(t)}, \ldots, \mathbf{v}_N^{(t)}]^T$. Substituting (9) into (8) and using the rotation given in (1) to restore data-closeness we can obtain the normal at iteration $(t+1)$ using the update equation $\mathbf{n}_p^{(t+1)} = \Theta_p \mathrm{Exp}_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{P}^T\mathbf{v}^{(t)})_p\right)$.

We may also exploit the constraint provided by our model to choose a better initialisation. We choose to initialise each normal to the point on their reflectance cone closest to the local average normal direction given by the intrinsic mean at that position. This is equivalent to rotating the local average normal onto the closest position on the reflectance cone, $\mathbf{n}_p^{(0)} = \Theta_p \boldsymbol{\mu}_p$. In practice, we find that this initialisation offers improved performance over the negative gradient initialisation described above.

The principal geodesic directions can be used to fit the statistical model for surface normal directions to image brightness data using the following algorithm.

Algorithm 1

1. *Initialise:* $\mathbf{n}_p^{(0)} = \Theta_p \boldsymbol{\mu}_p$*, set iteration* $t = 0$.
2. *Compute parameter vector:* $\mathbf{b}^{(t)} = \mathbf{P}^T\mathbf{v}^{(t)}$.
3. *Update normals:* $\mathbf{n}_p^{(t+1)} = \Theta_p Exp_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{b}^{(t)})_p\right)$*, set* $t = t + 1$.
4. *Stop if* $t >$*maximum number of iterations. Otherwise return to 2.*

The method hence combines a strict global constraint (projection onto the statistical model) with a hard local constraint (satisfaction of the image irradiance equation). We find the algorithm converges rapidly and offers stable performance on real world images. However, a number of obstacles are encountered when this simple approach is applied to real images. The simple reflectance model given Lambert's law assumes constant albedo and ignores the effect of cast shadows (regions in which the light source is intercepted by another part of the surface). The result is that the fit to the statistical model (9) is subject to a systematic error and becomes increasingly inaccurate when regions of low albedo dominate (for instance in the presence of facial hair) or when cast shadows become significant (as the light source direction is more extreme). In this case, a significant portion of the face may be in shadow and fitting the statistical model globally results in erroneous shape parameter estimates.

5 Robust Statistics

It is therefore clear that we require a more robust means to fit our statistical model to a potentially noisy observed field of normals, $\mathbf{n}$. The quality of this fit can be measured by calculating the distance between the observed and fitted normals on the plane. If $\mathbf{b}$ is the estimated parameter vector, the residual at point p is given by $\eta_p = \|\mathrm{Log}_{\boldsymbol{\mu}_p}(\mathbf{n}_p) - (\mathbf{P}\mathbf{b})_p\|$. The standard least squares fit given above, minimises the quantity:

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \eta_i^2.$$

This approach is unstable in the presence of outlying data, such as normals erroneously estimated from regions of low albedo or in cast shadow. In particular, the effect of outliers is to severely distort the estimated facial shape.

In this paper, we turn to the apparatus of robust statistics to help overcome this problem. M-estimators (maximum likelihood type estimators) aim to reduce the effect of outliers by replacing the squared residuals η_i^2 by a kernel function that limits the effects large residuals:

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} \sum_{i=1}^{N} \rho_\sigma(\eta_i), \tag{10}$$

where ρ is a robust kernel with width parameter σ.

The influence of a residual on the parameter estimate under a given M-estimator can be studied by examining its *influence function*, ψ_σ. This is the derivative of the error kernel:

$$\psi_\sigma(\eta_p) = \frac{\partial \rho_\sigma(\eta_p)}{\partial \eta_p}. \tag{11}$$

Hence, in the least squares case where $\rho_\sigma(\eta) = \eta^2$, the influence of a datum is $\psi_\sigma(\eta) = 2\eta$ and therefore increases linearly with the size the error. This is the source of the lack of robustness in least-squares estimation.

We propose a robust solution to (10) using a simple one-step weighted least squares approximation. To do so, we make use of the *weight function*, w_σ, which is related to the influence function by:

$$w_\sigma(\eta_p) = \frac{\psi_\sigma(\eta_p)}{\eta_p}. \tag{12}$$

The standard least-squares estimator applies a constant weight to each datum. On the other hand, an error kernel such as Huber's estimator [14] down-weights a datum once its residual exceeds σ:

$$\rho_\sigma(\eta) = \begin{cases} \eta^2 & \text{if } |\eta| < \sigma \\ 2\sigma|\eta| - \sigma^2 & \text{otherwise} \end{cases}, \qquad w_\sigma(\eta) = \begin{cases} 1 & \text{if } |\eta| < \sigma \\ \frac{\sigma}{|\eta|} & \text{otherwise} \end{cases}. \tag{13}$$

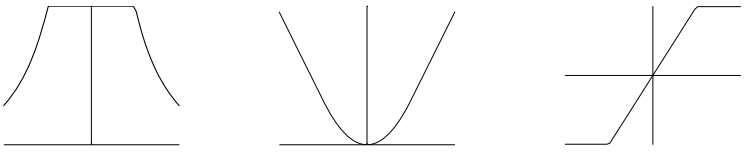
We show the weight function, error kernel and influence function for Huber's M-estimator in Fig 2. This is the M-estimator we use in the remainder of this paper.

We can incorporate the Huber weights into the least squares expression (9) by constructing a diagonal matrix of weights: $\mathbf{W} = \text{diag}\,(w_\sigma(\eta_1), \ldots, w_\sigma(\eta_N))$. Our one-step weighted least squares approximation of $\mathbf{b}$ is given by:

$$\mathbf{b}^{(t)} = C\mathbf{P}^T\mathbf{W}\mathbf{v}^{(t)}, \tag{14}$$

where C is a constant which compensates for the overall scaling effect of $\mathbf{W}$ on $\mathbf{b}$:

$$C = \sum_{p=1}^{N} \frac{1}{w_\sigma(\eta_p)} = N\text{Tr}(\mathbf{W}^{-1}). \tag{15}$$



$$w_\sigma(\eta) = \begin{cases} 1 \\ \frac{\sigma}{|\eta|} \end{cases} \qquad \rho_\sigma(\eta) = \begin{cases} \eta^2 \\ 2\sigma|\eta| - \sigma^2 \end{cases} \qquad \psi_\sigma(\eta) = \begin{cases} 2\eta \\ 2\sigma \text{Sign}(\eta) \end{cases}$$

Fig. 2. Huber's M-estimator

Computing (14) requires an initial estimate of $\mathbf{W}$ and hence the residuals. We therefore commence by calculating a least squares fit from which we can calculate the residuals η_p. This is equivalent to calculating (14) with an identity weight matrix, i.e. $\mathbf{W} = \mathbf{I}_N$. For a subsequent iteration t of the algorithm, we can use the weights calculated from the residuals at iteration $(t-1)$.

Implicit in the discussion above is that we have a means to estimate the standard deviation of the residual errors σ, which acts as the width parameter of the function ρ. A robust estimate of σ is required in order to distinguish outliers from inliers. For this reason, we use the *median absolute deviation* (MAD):

$$\text{MAD} = \text{median}\left(|\eta_p - \text{median}(\eta_p)|\right), p = 1 \ldots N, \tag{16}$$

which is related to the standard deviation by $\sigma = 1.4826 \times \text{MAD}$.

We can now outline the robust version of our algorithm, in which a weighted fit of the surface normal model is used to update the estimated normal directions.

Algorithm 2

1. *Initialise:* $\mathbf{n}_p^{(0)} = \Theta_p \boldsymbol{\mu}_p$, *set iteration* $t = 0$.
2. *Initialise weights:* $\mathbf{W}^{(0)} = \mathbf{I}_N$.
3. *Calculate robust estimate of parameter vector:* $\mathbf{b}^{(t)} = C\mathbf{P}^T\mathbf{W}^{(t)}\mathbf{v}^{(t)}$.
4. *Calculate residuals:* $\eta_p^{(t)} = \| Log_{\boldsymbol{\mu}_p}(\mathbf{n}_p^{(t)}) - (\mathbf{P}\mathbf{b}^{(t)})_p \|$
5. *Calculate weight matrix:* $\mathbf{W}^{(t+1)} = diag\left(w_\sigma^{(t)}(\eta_1), \ldots, w_\sigma^{(t)}(\eta_N)\right)$.
6. *Update normals:* $\mathbf{n}_p^{(t+1)} = \Theta_p Exp_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{b}^{(t)})_p\right)$, *set* $t = t + 1$.
7. *Stop if* $t >$*maximum number of iterations. Otherwise return to 3.*

It is clear that our algorithm shares much in common with the iteratively reweighted least-squares algorithm, in that the weights are calculated using the residuals from the previous iteration. The difference is that in step 6 we adjust the data to which the model is being fitted. The result of this is that during earlier iterations when the majority of residuals are large and the weight function serves only to coarsely mask unreliable areas, convergence is driven primarily by the hard irradiance constraint. During later iterations as reliable pixels contain

accurate estimates of the normal direction, the adjustment made in step 6 is minimal and therefore the residuals are small. Hence, the effects of the weighting becomes more subtle and the parameter estimate converges towards a robust solution.

5.1 Combining and Classifying

Upon convergence, if a pixel p has weight $w_p^{final} \approx 1$, this indicates a high confidence that the normal $\mathbf{n}_p^{final}$ is reliable. However, as the weight tends to 0, the normal $\mathbf{n}_p^{final}$ is likely to be erroneous due to violation of the assumptions of Lambert's law, e.g. non-constant albedo or lying in a cast shadow region. In this case, a more accurate estimate of $\mathbf{n}_p^{final}$ is given by the robust fit of the model to the global field of normals, e.g. $\mathrm{Exp}_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{b}^{final})_p\right)$.

For this reason, our best estimate of the underlying shape of the face is a weighted combination, in which pixels with a low weight are given a higher proportion of the normal $\mathrm{Exp}_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{b}^{final})_p\right)$ and a lower proportion of the normal $\mathbf{n}_p^{final}$, vice-versa for pixels with a high weight. This gives:

$$\mathbf{n}_p^{combined} = \frac{\overline{\mathbf{n}}_p}{\|\overline{\mathbf{n}}_p\|}, \tag{17}$$

where

$$\overline{\mathbf{n}}_p = w_p^{final}\mathbf{n}_p^{final} + \left(1 - w_p^{final}\right)\mathrm{Exp}_{\boldsymbol{\mu}_p}\left((\mathbf{P}\mathbf{b}^{final})_p\right). \tag{18}$$

With the estimated facial shape to hand, we may now go further and distinguish between pixels of low albedo and those in cast shadow regions. We may recover the surface height z_p by applying a standard surface integration method [15] on the field of normals $\mathbf{n}^{combined}$. Using a simple ray-tracing algorithm, we can assign a binary cast-shadow map:

$$\mathrm{shadow}(z_p, \mathbf{s}) = \begin{cases} 0 & \text{if pixel } p \text{ is in cast shadow} \\ 1 & \text{otherwise} \end{cases}. \tag{19}$$

For non-shadow regions, the albedo ω_p can be estimated by rearranging the image irradiance equation:

$$\omega_p = \frac{I_p}{\mathbf{n}_p^{combined}.\mathbf{s}}. \tag{20}$$

6 Experimental Results

In this section we present experiments using the technique described above. We begin by applying the method to known ground truth data allowing us to quantitatively asses the performance of the approach. We then apply the method to real world images, demonstrating the robustness of the approach under real world conditions.

We train our statistical model by applying PGA to a sample of 100 facial needle-maps. The data is acquired from the 3DFS dataset [16] which consists of 100 high resolution scans of subjects in a neutral expression. The scans were collected using a *Cyberware*$^{\text{TM}}$ 3030PS laser scanner. The database is pre-aligned, registration being performed using the optical flow correspondence algorithm of Blanz and Vetter [6]. For ground truth, we use a leave-one-out strategy in which we train the model with 99 sets of data, leaving the remaining needle-map as out-of-sample ground truth.

For real world images, we show reconstructions and reilluminations of images from the Yale-B database [2]. These contain albedo variation and cast shadows.

6.1 Ground Truth Data

In Fig. 3 we demonstrate the performance of our method on ground truth data. We apply our algorithm to a selection of images of rendered ground truth needle-maps including cast shadows. In column 1 we show the input images. The needle-maps of the out-of-sample subjects are rendered with Lambertian reflectance and a point light source with direction $\mathbf{s} = (-1, 0, 1)$, i.e. 45° from the viewing direction. We also simulate the effect of cast shadows using the shadow map shown in column 2. shadow$(z, \mathbf{s})$ is calculated from ground truth depth data. In column 3 we show the weight function $w_\sigma(\eta_p)$ for each pixel. It is clear that regions in cast shadow have been successfully down-weighted. In column 4 we show the needle-map $\mathbf{n}^{combined}$ calculated from the input image, rendered with frontal illumination. For comparison, in column 5 we show the ground truth needle-map similarly illuminated. There is a good agreement between the two, even in areas in which no information was present in the input image (i.e. those in cast shadows). This suggests that the robust fit of the model has recovered globally accurate shape information, and has filled-in the shadowed areas of the face. The mean surface normal error was typically $< 8°$ across the whole needle-map. Finally in column 6 we show the shadow map shadow$(z^{combined}, \mathbf{s})$, where $z^{combined}$ is the height map integrated from $\mathbf{n}^{combined}$. Again, there is a good agreement between columns 2 and 6, suggesting that this represents a viable means to estimate regions which are in cast shadow.

In Fig. 4 we examine the influence of illumination direction on the accuracy of the recovered surface normals. Once again we render ground truth, out-of-sample needle-maps with Lambertian reflectance and a point light source for which we simulate cast shadows. The direction of the light source is varied through a horizontal and vertical arc, i.e. from left to right and from top to bottom. Fig. 4 plots the average surface normal error against the angle between the light source and viewing direction. The arc from left to right $(\mathbf{s} = (-1, 0, 0), \ldots, (1, 0, 0))$ is shown as a solid line, while the arc from top to bottom $(\mathbf{s} = (0, 1, 0), \ldots, (0, -1, 0))$ is shown as a broken line. The plot demonstrates that our method recovers globally accurate shape information, even when the lighting direction is extreme and hence, much of the face is in shadow. For example, illumination from the extreme right or left still results in an average normal error of less than 10°.

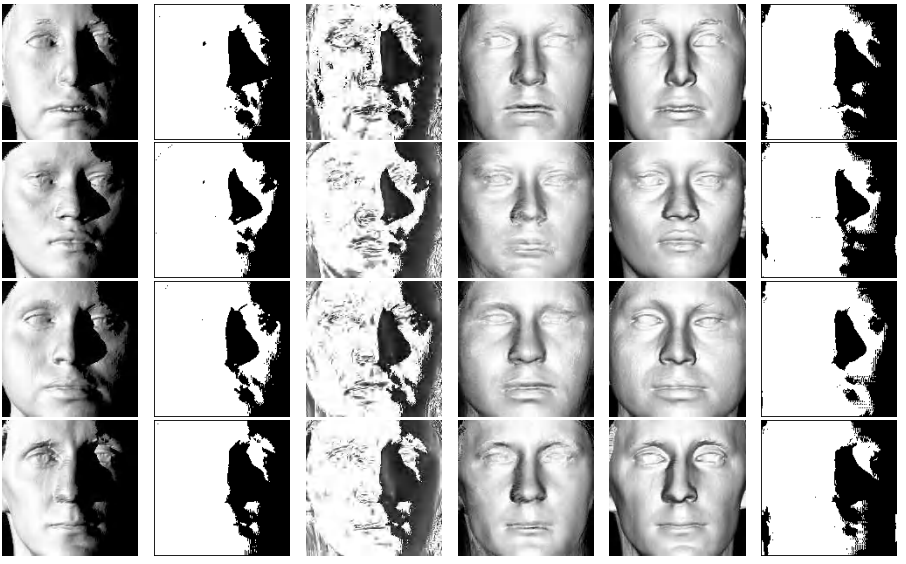


Fig. 3. Fitting to images of ground truth needle-maps rendered with Lambertian reflectance and cast shadows

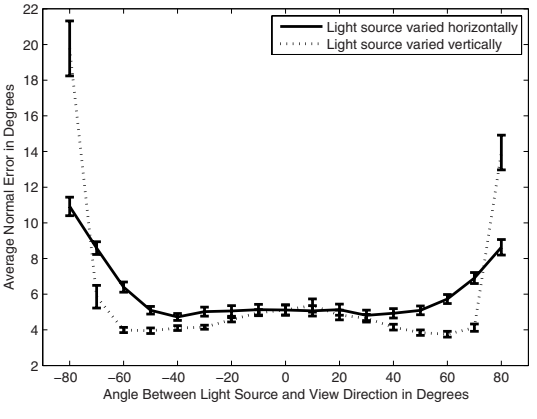


Fig. 4. Average angular error of the recovered normal versus illumination direction

6.2 Real World Data

In Fig. 5, we demonstrate the quality of the shape information our method can recover from real world images. From the input images in the top row, we use our method to estimate the needle-map $\mathbf{n}^{combined}$ and use the surface recovery method of Frankot and Chellappa to integrate the normals into a surface. We show these surfaces rendered with the estimated albedo and Lambertian

Fig. 5. Novel viewpoint of the surface recovered from the input image in the top row rotated 45° about the vertical axis

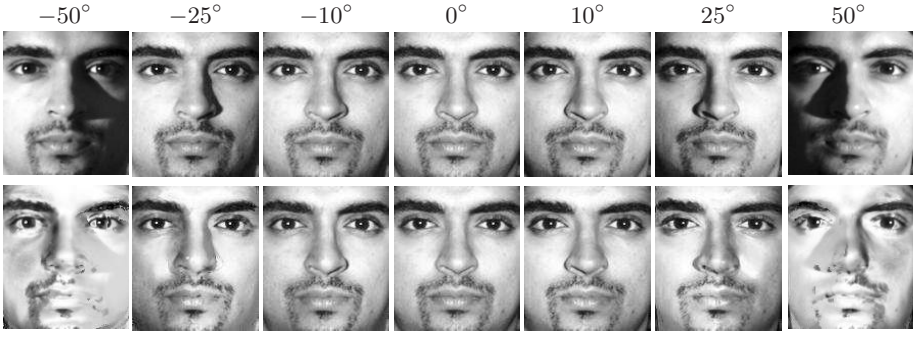


Fig. 6. Correcting for variations in illumination for real world images

reflectance, rotated 45° about the vertical axis. The images show considerable stability under large change in viewpoint.

In Fig. 6, we demonstrate the method of real images which contain cast shadows as well as albedo variations. The first row shows the input images of a single subject under varying illumination. The subject is a challenging choice due to the large albedo variations caused by facial hair. The light source is moved in an arc along the horizontal axis to subtend an angle of −50°, −25°, −10°, 0°, 10°, 25° and 50° with the viewing direction. We use our method to estimate the normals, albedo and shadow map. We use facial symmetry to fill-in the missing albedo values for the shadow regions. In the second row we show the estimated cast shadow map. Here, the cast shadows caused by the nose seem to correspond well with the input images. Finally in the third row, we show the recovered needle-maps rendered with the estimated albedo and frontal lighting, effectively correcting for variation in input lighting. These synthesised images

are of a good quality, even under large changes in illumination and manage to remove much of the effect of the cast shadows.

7 Conclusions

We have presented a statistical shape-from-shading framework in which the principal geodesics learnt from an exemplar distribution of surface normals provide a global constraint which we use to update the surface normal estimates. We fit the statistical model globally, but use robust statistics to ensure that regions of low albedo or which fall into a cast shadow have little or no impact on the parameter estimate. The technique is capable of recovering a useful estimate of facial shape, even when significant portions of the face are entirely in shadow.

We used the technique for a number of face analysis tasks. We began by examining the accuracy of the method on data with known ground truth. We then showed how a cast shadow map can be estimated from both ground truth and real world imagery. This may prove useful to further analysis of the face using other methods. Finally, we demonstrated that the facial shape estimated by our method, combined with an estimate of the albedo and shadow map, is sufficient to synthesise images under frontal lighting from extreme input lighting directions.

References

1. Zhao, W.Y., Chellappa, R.: Illumination-insensitive face recognition using symmetric SFS. In: Proc. CVPR, pp. 286–293 (2000)
2. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. 23, 643–660 (2001)
3. Prados, E., Faugeras, O.D.: Unifying approaches and removing unrealistic assumptions in shape from shading: Mathematics can help. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 141–154. Springer, Heidelberg (2004)
4. Dovgard, R., Basri, R.: Statistical symmetric shape from shading for 3D structure recovery of faces. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3022, pp. 99–113. Springer, Heidelberg (2004)
5. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proc. SIGGRAPH, pp. 187–194 (1999)
6. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE Trans. Pattern Anal. Mach. Intell. 25, 1063–1074 (2003)
7. Atick, J.J., Griffin, P.A., Redlich, A.N.: Statistical approach to SFS: Reconstruction of 3D face surfaces from single 2D images. Neural Comp. 8, 1321–1340 (1996)
8. Smith, W., Hancock, E.R.: Recovering facial shape using a statistical surface normal model. In: Proc. ICIP (2005)
9. Blanz, V., Basso, C., Poggio, T., Vetter, T.: Reanimating faces in images and video. In: Proce. EUROGRAPHICS, pp. 641–650 (2003)
10. Fletcher, P.T., Joshi, S., Lu, C., Pizer, S.M.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Trans. Med. Imaging 23, 995–1005 (2004)

11. Pennec, X.: Probabilities and statistics on riemannian manifolds: A geometric approach. Technical Report RR-5093, INRIA (2004)
12. Mardia, K.V., Jupp, P.E.: Directional Statistics. John Wiley and Sons Ltd, West Sussex (2000)
13. Fisher, N.I.: Spherical medians. J. R. Statist. Soc. B 47, 342–348 (1985)
14. Huber, P.: Robust Statistics. Wiley, Chichester (1981)
15. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 10, 439–451 (1988)
16. USF HumanID 3D Face Database, Courtesy of Sudeep. Sarkar, University of South Florida, Tampa, FL

Statistical Methods for Surface Integration

William A.P. Smith and Edwin R. Hancock

Department of Computer Science, The University of York, UK
{wsmith,erh}@cs.york.ac.uk

Abstract. In this paper we show how statistical constraints can be incorporated into the surface integration process. This problem aims to reconstruct the surface height function from a noisy field of surface normals. We propose two methods that employ a statistical model that captures variations in surface height. The first uses a coupled model that captures the variation in a training set of face surfaces in both the surface normal and surface height domain. The second is based on finding the parameters of a surface height model directly from a field of surface normals. We present experiments on ground truth face data and compare the results of the two methods with an existing surface integration technique.

1 Introduction

One of the simplest approaches to facial shape recovery using shape-from-shading is to extract a field of surface normals and then recover the surface height function by integrating the surface normals. In previous work [1] we have shown how a statistical model that captures the variation in the field of surface normals of a face can be incorporated into a shape-from-shading algorithm. This allows accurate surface orientation information to be recovered from single, real world images of faces. This $2\frac{1}{2}$D shape representation is useful in its own right for synthesising novel illuminations, performing recognition and estimating albedo maps.

For many purposes, the ultimate aim of facial shape-from-shading is to provide an estimate of the true 3-dimensional surface of the face. This is essential if we wish to synthesise images from a novel viewpoint or for model acquisition for applications such as animation. However, using existing surface integration techniques, the concave/convex ambiguities in the needle-map can lead to the distortion of the topography of the reconstructed face surface. One of the most serious instances of this problem is that the nose can become imploded [2].

1.1 Surface Integration

Suppose that a facial surface $F \in \mathbb{R}^3$ is projected orthographically onto the image plane and parameterised by the function $z(x, y)$. The second partial derivatives of this surface are independent of the order of differentiation:

$$\partial_{xy} z(x, y) = \partial_{yx} z(x, y). \tag{1}$$

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 427–441, 2007.
© Springer-Verlag Berlin Heidelberg 2007

If the surface gradients of a field of surface normals $\mathbf{n}(x,y)$ are $p(x,y) = \frac{n(x,y)_x}{n(x,y)_z}$ and $q(x,y) = \frac{n(x,y)_y}{n(x,y)_z}$, they are said to be *integrable* if they also satisfy this condition, i.e.

$$\partial_y p(x,y) = \partial_x q(x,y). \tag{2}$$

Recovering the surface height function from a field of surface normals or surface gradient estimates is known as the *surface integration* or *height-from-gradient* problem. Given a field of surface normals that satisfy the integrability constraint, this problem is straightforward to solve. It simply requires the assignment of an initial height to a chosen starting point, an arbitrary path of integration to be followed through the field of normals and the height to be summed from the surface gradients given by the local surface normal. However, difficulties arise when the surface normals are noisy or contain systematic errors. In this case, the resulting surface will depend on the path of integration chosen.

1.2 Related Work

This problem has received considerable attention in its own right. Typically it is posed in either local or global terms. Local integration methods [3] are conceptually simpler but are highly sensitive to noise and are hence unsuitable for use with normals estimated using shape-from-shading. Global approaches tend to be more robust to noise and recover smoother surfaces. Perhaps the most popular global method is that of Frankot and Chellappa [4]. They project the estimated gradient field into the Fourier domain to impose integrability constraints and then recover surface height using an inverse Fourier transform. Although this approach is both elegant and numerically efficient it is sensitive to abrupt changes in orientation, for example at the object boundary. This means that careful parameter selection is necessary to recover a stable surface. It also requires that the surface be continuous. Wei and Klette [5] have enhanced this approach by incorporating more complex regularisation constraints in the Fourier domain. More recently, Agrawal et al. [6] have revisited this problem and shown how to pose it algebraically. They exclude gradient measurements which do not have zero curl, i.e. where the surface is not locally integrable, and provide improved results over the original method.

Although these methods are useful, they introduce errors of their own and when combined with imperfect surface normals provided by shape-from-shading, the two sources of error compound each other. Figure 1 highlights these problems. On the left of the figure a depth map of a face is shown, which was recovered using a laser range sensor. The second column shows a profile view of the depth map rendered with Lambertian reflectance and a frontal light source. The third column shows the result of using shape-from-shading to recover a field of surface normals from an image of the face in a frontal view and integrating these normals into a surface (we use the Worthington and Hancock [7] shape-from-shading algorithm and Frankot and Chellappa [4] surface integration algorithm). The errors introduced by both processes are compounded resulting in a very inaccurate surface. To illustrate this, the profile view of the surface recovered from ground

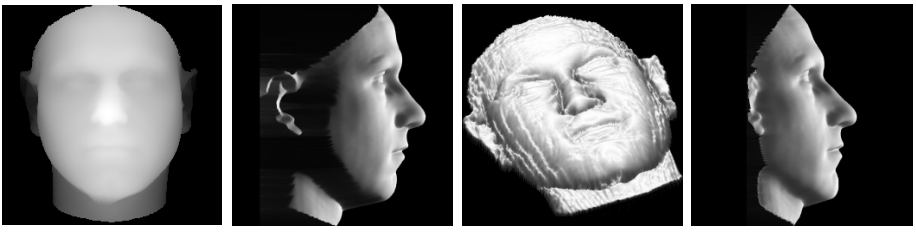


Fig. 1. Errors compound each other when shape-from-shading is combined with surface integration. From left to right: depth map, ground truth profile view, surface recovered using shape-from-shading, profile view of surface integrated from ground truth surface normals using Frankot and Chellappa's method.

truth surface normals using the method of Frankot and Chellappa [4] is shown on the right of the figure. Even with noise-free, ground truth normals the recovered surface is inaccurate. Areas which were close to the occluding boundary in the front view have been poorly recovered resulting in a loss of depth (the ear is too close to the nose). Also, the steepness of the surface at the base of the nose has caused the height of the lower half of the face to be exaggerated (in the ground truth surface the chin is at almost the same depth as the forehead, whereas in the estimated surface the chin is noticeably further forward).

1.3 Exploiting a Statistical Model of Surface Height

The improvements in the surface orientation information that may be recovered using the methods described in our previous work [1], depend upon imposing a global statistical constraint on the field of surface normals. It seems clear that applying a generic surface integration technique to these normals ignores the important information we have recovered in fitting this statistical model. In other words, the resulting surface height function is free to assume any global shape, without being subject to a similar model-based constraint. Our aim is therefore to show how a shape model may be incorporated into the surface integration process. In this paper we describe two ways in which this may be accomplished.

A Coupled Model. We begin by showing how a coupled statistical model may be built that jointly describes variations in surface normal direction and height over the surface of a face. We construct separate eigenspaces for the surface normal and height variations from the covariance matrices of the training data. With the parameters of the surface normal subspace to hand, we find the parameters of the coupled model that give the best match to those in the surface normal subspace. Hence, the model is fitted to needle-map data in the surface-normal subspace, and the recovered parameters are used to project the fitted model into the height subspace to recover the facial surface. This process implicitly recovers the surface height function without having to explicitly integrate the recovered field of surface normals.

This somewhat naive approach relies entirely on the statistical learning method employed (namely PCA) to explain the relationship between the field of surface normals and the surface height function. Nonetheless, we show that the approach provides improved results over using a generic surface integration method.

Imposing Height Constraints on Surface Normals. We then take this analysis one step further by discarding the statistical model of surface normals altogether. Instead we show that a statistical surface height model can be used to provide a similar constraint for surface normals. We do so by examining the relationship between the parameters of a surface height model and the surface normals of the resulting surface. We first show how a field of surface normals may be expressed in terms of the parameters of a height model. We demonstrate that by minimising the error between a field of normals expressed in these terms and an input field of normals, we find the closest field of normals in which both integrability and a statistical constraint on the corresponding surface are enforced. We term this process 'model-based integration'. Once again, the method yields a height map without having to explicitly integrate the surface normals.

2 Coupling Height and Surface Normal Variation

We assume a statistical model for fields of surface normals has been constructed according to the method described in [1]. Here, PCA is used to find the orthogonal modes of variation in a training sample of K fields of surface normals. The matrix containing the K eigenvectors of surface normal variation are captured in $\mathbf{P}_s$. A field of $N = X_{\text{res}} \times Y_{\text{res}}$ surface normals may be represented by the matrix $\mathbf{U} \in \mathbb{R}^{N\times 2}$ which contains the coordinates of each surface normal under azimuthal equidistant projection (see [1] for the details of this projection). The model may then be used to describe this field of surface normals using a low-dimensional parameter vector:

$$\mathbf{b}_s = \mathbf{P}_s^T \mathbf{U}. \tag{3}$$

We now describe how the statistical model of variation in surface height is constructed.

2.1 A Surface Height Model

The training set comprises K range images of size $N = X_{\text{res}} \times Y_{\text{res}}$ pixels. The kth training sample may be represented by the vector of height values ordered according to the raster scan

$$\mathbf{z}^k = [z^k(1,1), z^k(1,2), \ldots, z^k(X_{\text{res}}, Y_{\text{res}})]^T. \tag{4}$$

The mean surface height vector $\hat{\mathbf{z}}$ is given by $\hat{\mathbf{z}} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{z}^i$. We form the $N \times K$ data matrix of height values using:

$$\mathbf{D}_h = [(\mathbf{z}^1 - \hat{\mathbf{z}})|(\mathbf{z}^2 - \hat{\mathbf{z}})| \ldots |(\mathbf{z}^K - \hat{\mathbf{z}})]. \tag{5}$$

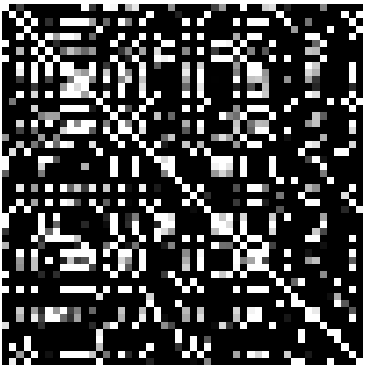


Fig. 2. Correlation matrix for the first 50 parameters of the surface normal and height models

In precisely the same manner as for the surface normals, we once again use PCA to extract the set of K orthogonal modes of variation $\mathbf{P}_h$ from the covariance matrix $\mathbf{L}_h = \frac{1}{K}\mathbf{D}_h\mathbf{D}_h^T$. Again, a long-vector of height values $\mathbf{z}^k$ can be projected onto the eigenvectors and represented using the vector of model parameters $\mathbf{b}_h = \mathbf{P}_h^T(\mathbf{z}^k - \hat{\mathbf{z}})$.

2.2 Coupling the Surface Normal and Height Models

Each training sample can be summarised by the parameter vectors $\mathbf{b}_s$ and $\mathbf{b}_h$, representing the needle-map and height map of the sample respectively. Since surface height and the field of surface normals are closely related (recall that $\mathbf{n}(x,y) = (\partial_x z(x,y), \partial_y z(x,y), -1)$) the two sets of parameters will contain strong correlations. To illustrate this, Figure 2 shows the correlation matrix for the first 50 parameters of the surface normal and height models. There are strong correlations between the two, particularly along the diagonal. This suggests that modes of corresponding importance capture similar modes of variation. Intuitively, our argument is that a similar combination of modes of variation will describe a face's shape in both the surface normal and surface height domain. We intend to learn this relationship using PCA.

In both models, we may consider small scale variation as noise. Hence, if the ith eigenvalue for the surface normal model is $\lambda_{s,i}$, we need only retain S eigenmodes to retain c percent of the model variance. We choose S as the smallest integer that satisfies $\sum_{i=1}^{S} \lambda_{s,i} \geq \frac{c}{100} \sum_{i=1}^{K} \lambda_{s,i}$. Similarly for the height model we retain H eigenmodes to capture c percent of the variance.

For the kth training sample we can generate the concatenated vector of length $S + H$:

$$\mathbf{b}^k = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s^k \\ \mathbf{b}_h^k \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \mathbf{P}_s^T \mathbf{U}^k \\ \mathbf{P}_h^T(\mathbf{z}^k - \hat{\mathbf{z}}) \end{pmatrix}, \tag{6}$$

where $\mathbf{W}_s$ is a diagonal matrix of weights for each surface normal model parameter, allowing for the different relative weighting of the surface normal and depth models. The reason for performing this weighting is that the elements of $\mathbf{b}_s$ have units of radians, $\mathbf{b}_h$ have units of distance, so they cannot be compared directly. We follow Cootes and Taylor [8] and set $\mathbf{W}_s = r\mathbf{I}_S$ where r^2 is the ratio of the total depth variance to the total surface normal variance, i.e.

$$r = \sqrt{\frac{\sum_{i=1}^{H} \lambda_{h,i}}{\sum_{i=1}^{S} \lambda_{s,i}}}, \tag{7}$$

and $\mathbf{I}_S$ is the identity matrix of size S. The coupled model data matrix is $\mathbf{D}_c = [\mathbf{b}^1|\dots|\mathbf{b}^K]$, where $\mathbf{b}^k$ represents the concatenated vector for the kth training sample. Because of the choice of weight matrix, the matrix $\mathbf{D}_c$ is the correlation matrix for the concatenated parameters. In order to find redundancies in the correlations between the two models, we now apply a final PCA to give the coupled model:

$$\mathbf{b} = \mathbf{P}_c\mathbf{c} = \begin{pmatrix} \mathbf{P}_{cs} \\ \mathbf{P}_{ch} \end{pmatrix} \mathbf{c}, \tag{8}$$

where $\mathbf{P}_c$ are the eigenvectors and $\mathbf{c}$ is a vector of coupled parameters controlling both the surface normal model and height model simultaneously. The matrix $\mathbf{P}_{cs}$ has S rows, and represents the first S eigenvectors, corresponding to the surface normal subspace of the model. The matrix $\mathbf{P}_{ch}$ has H rows, and represents the final H eigenvectors, corresponding to the height subspace of the model.

Crucially, we may now express the vectors of projected surface normal directions and height values directly in terms of the parameter vector $\mathbf{c}$:

$$\mathbf{U} = \mathbf{P}_s\mathbf{W}_s^{-1}\mathbf{P}_{cs}\mathbf{c} \tag{9}$$

$$\mathbf{z} = \hat{\mathbf{z}} + \mathbf{P}_h\mathbf{P}_{ch}\mathbf{c} \tag{10}$$

For compactness we write: $\mathbf{Q}_s = \mathbf{P}_s\mathbf{W}_s^{-1}\mathbf{P}_{cs}$ and $\mathbf{Q}_h = \mathbf{P}_h\mathbf{P}_{ch}$.

2.3 Fitting the Coupled Model to Surface Normal Data

Fitting the model to data involves estimating the parameter vector $\mathbf{c}$ from surface normal data. To do this we seek the coupled model parameters which minimise the error between the measured and reconstructed parameters describing the field of surface normals. In doing so we implicitly recover the surface which is also represented by the coupled model parameters.

Suppose that $\mathbf{U}$ is a vector of length $2N$ that represents a field of surface normals under azimuthal equidistant projection obtained by applying the shape-from-shading algorithm described in [1] to the brightness image of a face. We fit the model to data seeking the vector $\mathbf{c}^*$ of length $S + H$ that satisfies the condition

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \|\mathbf{P}_s^T\mathbf{U} - \mathbf{W}_s^{-1}\mathbf{P}_{cs}\mathbf{c}\|^2. \tag{11}$$

The corresponding best-fit vector of depth values is given by

$$\mathbf{z} = \hat{\mathbf{z}} + \mathbf{Q}_h \mathbf{c}^*. \tag{12}$$

This provides us with a means by which to recover surface height from a field of surface normals in a model-based way. In practise we use a Matlab implementation of a quasi-Newton minimisation process to solve Equation 11. In our experiments, we contrast this with using a generic height-from-gradient technique and show that the method is capable of recovering accurate profiles from frontal images.

3 Imposing Surface Height Constraints on Surface Normals

The method described in the previous section suffers from a number of drawbacks. The first is that recovering the best-fit coupled model parameters requires the solution of a minimisation problem which must be solved iteratively. The second is that the coupled model relies on learning a linear relationship between the parameters of a statistical surface normal and surface height model. This relationship is in fact more complex and non-linear. Finally, the coupled model approach requires both a surface normal and surface height model. As we will show in our experimental section, the surface normal model is less efficient in terms of the number of dimensions that need to be retained.

In this section we take a step further than in the previous section, by showing that in fact we can impose the constraints of a surface height model in the surface normal domain.

3.1 A Global Statistical Integrability Constraint

We can express the facial surface $F \in \mathbb{R}^3$ projected orthographically onto the image plane and parameterised by the function $z(x, y)$ in terms of a base surface $\hat{z}(x, y)$ plus a linear combination of K surface functions Ψ_i (or modes of variation):

$$z_{\mathbf{b}}(x, y) = \hat{z}(x, y) + \sum_{i=1}^{K} b_i \Psi_i(x, y), \tag{13}$$

where the coefficients b_i are the surface parameters. Two instances of this representation have proven useful to the shape-from-shading problem. The first is that of Frankot and Chellappa [4], where the fourier basis functions are used. These provide a complete orthonormal basis for z. However, the power of representation of the fourier basis means that they form a rather weak constraint on the surface. The second instance in which this formulation has been used is where the basis functions form a statistical model learnt from exemplar facial surfaces. This is the model used in the previous section where the base shape and modes of variation are given by applying PCA to a representative sample of

exemplar surfaces. The base surface $\hat{z}$ is the mean average of the training samples and Ψ_i is the eigenvector of the covariance matrix of the training samples corresponding to the ith largest eigenvalue. It is this model that we use in the following sections.

3.2 Normals from Height Parameters

We may express the normals of the surface directly in terms of the parameter vector $\mathbf{b}$:

$$\mathbf{n_b}(x,y) = \begin{pmatrix} \partial_x \hat{z}(x,y) + \sum_{i=1}^{K} b_i \partial_x \Psi_i(x,y) \\ \partial_y \hat{z}(x,y) + \sum_{i=1}^{K} b_i \partial_y \Psi_i(x,y) \\ -1 \end{pmatrix}. \tag{14}$$

In other words, the normal may be expressed in terms of the sum of the gradient of the mean surface with a linear combination of the gradients of the eigenvectors at the corresponding point. If we write the partial derivatives of the mean surface as: $\hat{p}(x,y) = \partial_x \hat{z}(x,y)$ and $\hat{q}(x,y) = \partial_y \hat{z}(x,y)$ and define $p_i(x,y) = \partial_x \Psi_i(x,y)$ and $q_i(x,y) = \partial_y \Psi_i(x,y)$, i.e. the partial derivatives of the eigenvectors in the x and y directions, we may rewrite this as:

$$\mathbf{n_b}(x,y) = \begin{pmatrix} \hat{p}(x,y) + \sum_{i=1}^{K} b_i p_i(x,y) \\ \hat{q}(x,y) + \sum_{i=1}^{K} b_i q_i(x,y) \\ -1 \end{pmatrix}. \tag{15}$$

When we wish to refer to the corresponding vectors of unit length we use: $\hat{\mathbf{n}}_\mathbf{b}(x,y) = \frac{\mathbf{n_b}(x,y)}{\|\mathbf{n_b}(x,y)\|}$.

A field of normals expressed in this manner satisfies two constraints. First, the field of normals will be integrable since they correspond exactly to the surface given by Equation 13. Second, the surface corresponding to the field of normals is also constrained to lie within the span of the surface height model. We term this constraint *model-based integrability*.

3.3 Model-Based Integration

To apply the constraint to a (possibly non-integrable) field of surface normals $\mathbf{n}(x,y)$, we seek the parameter vector $\mathbf{b}^*$, whose field of surface normals given by Equation 15 minimises the distance to $\mathbf{n}(x,y)$. We pose this as minimising the squared error between the surface gradients of $\mathbf{n}(x,y)$ and those given by Equation 15. The surface gradients of $\mathbf{n}(x,y)$ are $p(x,y) = \frac{n_x(x,y)}{n_z(x,y)}$ and $q(x,y) = \frac{n_y(x,y)}{n_z(x,y)}$. The optimal solution is therefore given by:

$$\begin{aligned} \mathbf{b}^* = \arg\min_{\mathbf{b}} \sum_{x,y} & \left[\hat{p}(x,y) + \sum_{i=1}^{K} b_i p_i(x,y) - p(x,y)\right]^2 \\ & + \left[\hat{q}(x,y) + \sum_{i=1}^{K} b_i q_i(x,y) - q(x,y)\right]^2 \end{aligned} \tag{16}$$

The solution to this minimisation is linear in $\mathbf{b}$ and is solved using linear least squares as follows. If the input image is of dimension $N = X_{\text{res}} \times Y_{\text{res}}$, we

subtract the mean surface gradients from the surface gradients of the field of surface normals $\mathbf{n}(x, y)$ to form a centred vector of length $2N$:

$$\mathbf{G} = \begin{bmatrix} p(1,1) - \hat{p}(1,1) \\ q(1,1) - \hat{q}(1,1) \\ \cdots \\ p(X_{\text{res}}, Y_{\text{res}}) - \hat{p}(X_{\text{res}}, Y_{\text{res}}) \\ q(X_{\text{res}}, Y_{\text{res}}) - \hat{q}(X_{\text{res}}, Y_{\text{res}}) \end{bmatrix} \tag{17}$$

We then form the $2N \times K$ matrix of the surface gradients of the eigenvectors, $\mathbf{\Psi}$, whose ith column is $\mathbf{\Psi}_i = [p_i(1,1), q_i(1,1), \ldots, p_i(X_{\text{res}}, Y_{\text{res}}), q_i(X_{\text{res}}, Y_{\text{res}})]^T$. We may now state our least squares problem in terms of matrix operations:

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} \|\mathbf{\Psi}\mathbf{b} - \mathbf{G}\|^2. \tag{18}$$

The least squares solution is found using the pseudoinverse of $\mathbf{\Psi}$:

$$\mathbf{b}^* = \left(\mathbf{\Psi}^T\mathbf{\Psi}\right)^{-1}\mathbf{\Psi}^T\mathbf{G}. \tag{19}$$

This is a computationally efficient operation since it consists of only matrix multiplications. With the optimal parameter vector to hand, the field of surface normals satisfying the model-based integrability constraint is given by Equation 15. Furthermore, we have also implicitly recovered the surface height, which is given by Equation 13. This provides a means to recover the surface height from a field of normals without explicitly integrating the field of normals. To do so, we take as input a field of surface normals, fit the surface height model to this field of surface normals using Equation 19 and return the corresponding surface height given by Equation 13.

4 Experiments

In this section we present experimental results using the two methods described in this paper for surface height recovery.

We begin by comparing statistical models of surface height and surface normal direction. We then provide a thorough comparison of the surfaces recovered using the coupled model described in Section 2 and the model-based integration approach described in Section 3 using ground truth data.

We draw conclusions on which method appears to provide the best accuracy in terms of both the recovered surface height function.

4.1 Comparing Surface Normal and Height Models

We begin by comparing statistical models of facial shape constructed using surface normals and surface height. The models were trained on data from the 3DFS database [9]. Although trained on the same data and modelling the same underlying structures, the two models are in fact quite different. In Figure 3 we

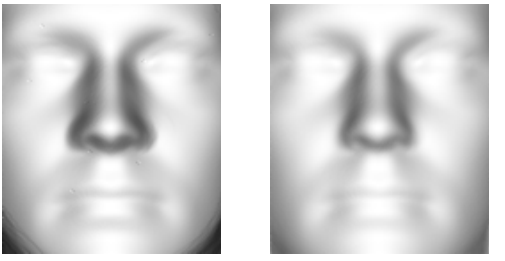


Fig. 3. Comparison of average face surface and average field of surface normals

show the average face surface on the left and the average field of surface normals on the right (we use the intrinsic mean definition of the average normal as in [10]). Both surface representations are rendered with Lambertian reflectance and frontal illumination. It is clear that the two show clear structural differences. This is because they are based on minimising different quantities. The average field of surface normals aims to minimise the difference in terms of surface orientation to all training samples while the average face surface aims to minimise the difference in terms of surface height.

We now examine the efficiency of the two facial surface models and the coupled model in terms of the cumulative variance captured. In Figure 4 we plot

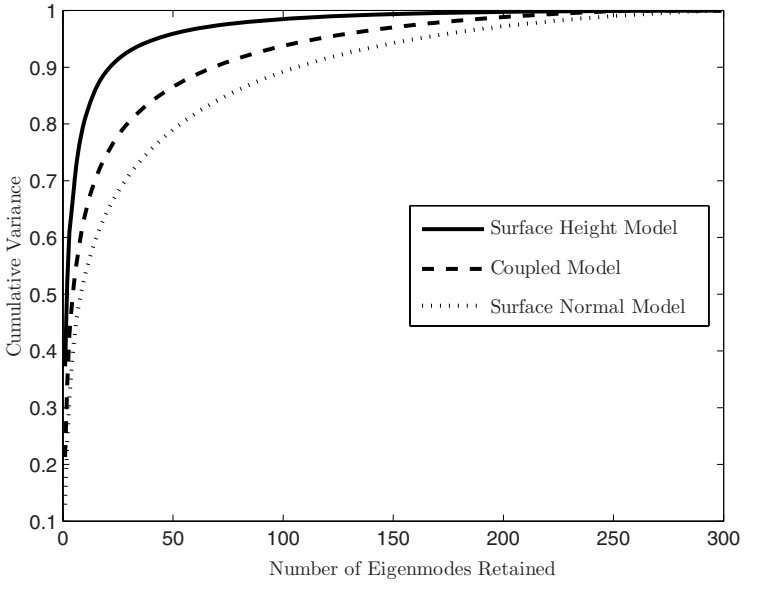


Fig. 4. Plot of cumulative variance versus number of eigenmodes used for surface height model (solid line), coupled model (dashed line) and surface normal model (dotted line)

cumulative variance against the number of eigenmodes retained. It is evident that fewer eigenmodes are required to capture variance in facial height than in facial needle maps. This is because the surface normal at each point has two degrees of freedom whereas the surface height value has only one. We retained 123 dimensions of the surface height model and 250 dimensions of the surface normal model (each accounting for 99% of the variance). As would be expected, the coupled model lies between the two and requires 207 dimensions to capture 99% of the variance.

4.2 Surface Integration

We now evaluate the two surface integration methods proposed in sections 2 and 3 on ground truth data. We use the generic surface integration technique of Frankot and Chellappa [4] to provide a comparison. We use a leave-one-out validation strategy, in which we train the statistical models with all but one of the facial surfaces. For this out-of-sample subject we compute the field of surface normals which we provide as input to each of the surface integration techniques. This is done as follows:

- **Coupled Model:** We fit the surface normal model to the input field of surface normals by calculating the best fit parameter vector. We then recover the coupled model parameters that best reconstruct this parameter vector using Equation 11. The estimated surface height is then given by Equation 12.
- **Model-based Integration:** We find the surface height model parameters that produce a surface whose surface normals best reconstruct the input field of surface normals by solving Equation 19. The estimated surface height is then given by Equation 13.

For each of 200 subjects we calculate the root mean square (RMS) error between the estimated and ground truth surfaces for the two proposed approaches and the result of applying the Frankot and Chellappa [4] algorithm. In Figure 5 we show the RMS errors (in centimetres) for each subject ordered from the best- to worst-case. The results of the Frankot and Chellappa algorithm are shown by the blue dot-dashed line. The results of using the coupled model are shown by the green dotted line and the results using model-based integration are shown by the red dashed line. We also plot the error of projecting the input ground truth surface onto the surface height model. This is the minimum error with which the surface height model can represent the input surface. This represents a baseline error which neither of the proposed methods could outperform.

The first feature to note is that both of the proposed methods substantially outperform the Frankot and Chellappa algorithm. The average RMS error across all subjects for the Frankot and Chellappa method is 0.4749 cm. The coupled model and model-based integration approaches both offer improved performance with model-based integration outperforming the coupled model. The average RMS error across all subjects for the coupled model and model-based integration are 0.2244 cm and 0.1367 cm respectively. For comparison the average RMS error

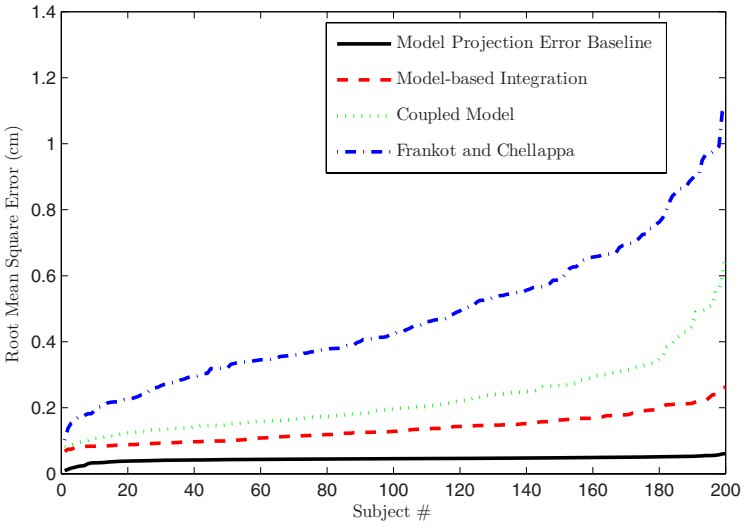


Fig. 5. RMS surface height errors for 200 subjects using 3 surface integration techniques applied to ground truth normals

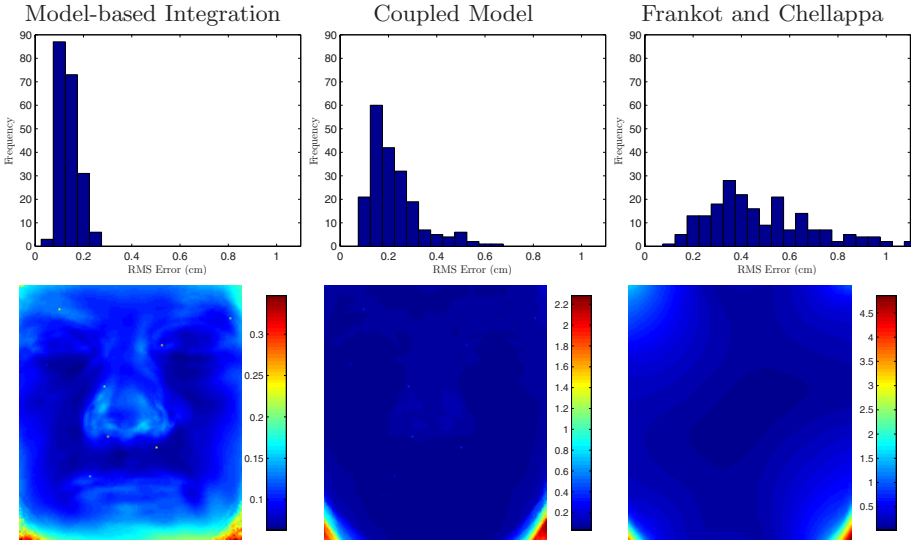


Fig. 6. Surface integration results: frequency histograms of RMS surface height errors and error maps of average RMS surface height errors at each pixel

across all subjects for the baseline model projection is 0.0448 cm. In Figure 6 we present alternative visualisations of these errors. In the top row we show frequency histograms showing the distribution of RMS errors across the 200

subjects for each of the three methods. In the second row we show error maps in which the colour represents the average RMS error at that pixel (note that different scales are used for each plot, the units are centimetres). These plots highlight the spatial distribution of errors for the three methods. In Table 1

Table 1. Comparing the accuracy of different surface integration methods. Values shown are average RMS errors in centimetres over 200 samples. The parenthesised values are the standard errors of the RMS errors.

Surface Integration from Ground Truth Normals		
Frankot and Chellappa	Coupled Model	Model-based Integration
0.4749cm ($S_E = 0.0149$cm)	0.2244cm ($S_E = 0.0075$cm)	**0.1367**cm ($S_E = 0.0030$cm)

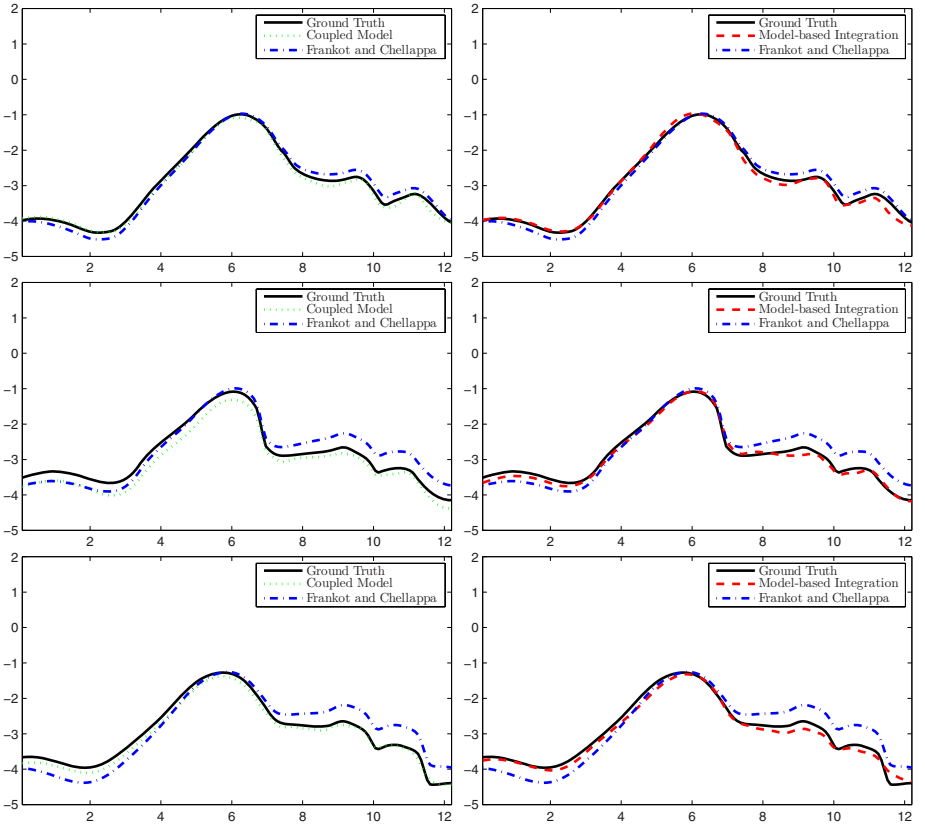


Fig. 7. Evaluating surface integration results based on contour of profile view of recovered surface. The reconstructed height along the central vertical scanline is shown. For three faces we compare results using the coupled model (left column) with results using model-based integration (right column). We also show ground truth and the result of the Frankot and Chellappa algorithm for comparison.

we summarise these results. We show the average RMS errors in centimetres of the recovered surface height function (the parenthesised values are the standard errors). The table highlights that both the methods developed in this paper offer an improvement over using the generic surface integration method of Frankot and Chellappa. Moreover, they show that model-based integration outperforms the coupled model.

In Figure 7 we show some specific examples from the above experiment. We compare contours of the profile view of the recovered surface (the central vertical scan-line) for 4 subjects. We show the ground truth contour as a solid black line and use the same line labelling convention as in Figure 5 for the estimated surfaces. The units of both axes are centimetres.

In the first example, all three methods perform well. The second example shows a case where model-based integration performs well, whilst the third example shows a case in which both model-based integration and the coupled model perform well. In these latter two cases the Frankot and Chellappa algorithm performs poorly. Here, the steepness of the surface at the base of the nose has caused the surface to be warped, leaving the lower half of the face estimated too high and the top half too low. This sensitivity to steeply inclined surface patches is a known weakness of the Frankot and Chellappa algorithm.

5 Conclusions

In this paper we have considered how statistical methods can be incorporated into a surface integration algorithm. We motivated the work by arguing that, if surface normal recovery can be improved using a statistical model [1], the surface integration process should be subject to a similar statistical constraint. For this reason we developed two surface integration methods which use a statistical model of surface height.

The first aims to learn the relationship between the parameters of a statistical surface normal model and surface height model based on a linear analysis of the correlation matrix between the two sets of parameters. To perform surface integration, we search for the parameters of this coupled model that best reconstruct the surface normal model parameters of the input field of surface normals.

The second method was based on a more thorough analysis of the relationship between the parameters of the surface height model and the resulting field of surface normals. We derive a method for recovering the surface height parameters directly from a field of surface normals, a process we term 'model-based integration'. The resulting algorithm has a number of desirable properties. Firstly, model-based integration is accomplished using matrix multiplications only whereas fitting the coupled model requires the solution of a minimisation problem. Secondly, the depth model is more efficient in terms of the number of dimensions that must be retained to preserve a given amount of variance. Further, half as many values are required to describe each dimension since only the surface height is being modeled as opposed to the two components of surface orientation. Finally, our experimental results show that for the

purposes of surface integration, the use of the model-based integrability constraint outperforms the coupled model when applied to ground truth data.

References

1. Smith, W.A.P., Hancock, E.R.: Recovering facial shape using a statistical model of surface normal direction. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1914–1930 (2006)
2. Castelán, M., Hancock, E.R.: Acquiring height data from a single image of a face using local shape indicators. Comput. Vis. Image Underst. 103, 64–79 (2006)
3. Wu, Z., Li, L.: A line integration based method for depth recovery from surface normals. Comput. Vis. Graph. Image Process 43, 53–66 (1988)
4. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 10, 439–451 (1988)
5. Wei, T., Klette, R.: Height from gradient using surface curvature and area constraints. In: Proc. ICVGIP, pp. 204–210 (2002)
6. Agrawal, A., Chellappa, R., Raskar, R.: An algebraic approach to surface reconstruction from gradient fields. In: Proc. ICCV 1, 174–181 (2005)
7. Worthington, P.L., Hancock, E.R.: New constraints on data-closeness and needle map consistency for shape-from-shading. IEEE Trans. Pattern Anal. Mach. Intell. 21, 1250–1267 (1999)
8. Cootes, T.F., Taylor, C.J.: Statistical models of appearance for computer vision. Technical report, Division of Imaging Science and Biomedical Engineering, University of Manchester (2001)
9. USF HumanID 3D Face Database, Courtesy of Sudeep. Sarkar, University of South Florida, Tampa, FL
10. Smith, W.A.P., Hancock, E.R.: Face recognition using 2.5d shape information. In: Proc. CVPR 2, 1407–1414 (2006)

Skeleton Surface Generation from Volumetric Models of Thin Plate Structures for Industrial Applications

Hiromasa Suzuki[1], Tomoyuki Fujimori[1], Takashi Michikawa[1], Yasuhiko Miwata[2], and Noriyuki Sadaoka[3]

[1] Research Center for Advanced Science and Technology, University of Tokyo, Japan
suzuki@den.rcast.u-tokyo.ac.jp
[2] Toyota Motor Corp., Japan
[3] Hitachi Ltd., Japan

Abstract. Computed Tomography (CT) is a powerful non-destructive measurement technology to generate cross-sectional X-ray images of objects, from which three-dimensional volumetric models can be constructed. In this study, we focus on industrial applications for X-ray CT in the analysis of thin plate structures, and propose in particular a method to generate sufficiently accurate skeleton meshes from a volumetric model of a thin plate structure. We use a geodesic-based skeletonization algorithm to extract skeleton cells, which can then be contoured to generate a skeleton surface. Since the thin plate structure has junctions, the skeleton surface is non-manifold around these areas. Because of this, the Marching Cubes algorithm is extended to enable handling of multiple labels (signs) to generate non-manifold meshes. The generated mesh is optimized to pass through the mid of the plate by interpolating mapping. Some experimental results for industrial samples are shown using our implemented prototype system, and it is proven that the accuracy of the generated mesh is sufficient for industrial applications.

1 Introduction

Computed Tomography (CT) is a powerful non-destructive measurement technology to generate cross-sectional X-ray images of objects. In this study, we focus on industrial applications for X-ray CT in the analysis of thin plate structures such as those shown in Fig. 1(a). It is assumed that these thin plate structures are smooth in shape and that their thickness also changes very gradually. In Fig. 1(a), each pixel has a grey value which is basically proportional to the material density. For instance, a pixel corresponding to iron sheet metal has a CT value of around 10,000 (a foreground pixel), while a pixel corresponding to air (a background pixel) has a value of around zero. By building up these sectional images, a *volumetric model* of *voxels* is created in a three-dimensional lattice arrangement (Fig. 1(c)). The size of the voxels is determined by the pixel size and the slice thickness. In our experiments, the pixel size is usually 0.4mm ×

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 442–464, 2007.
© Springer-Verlag Berlin Heidelberg 2007

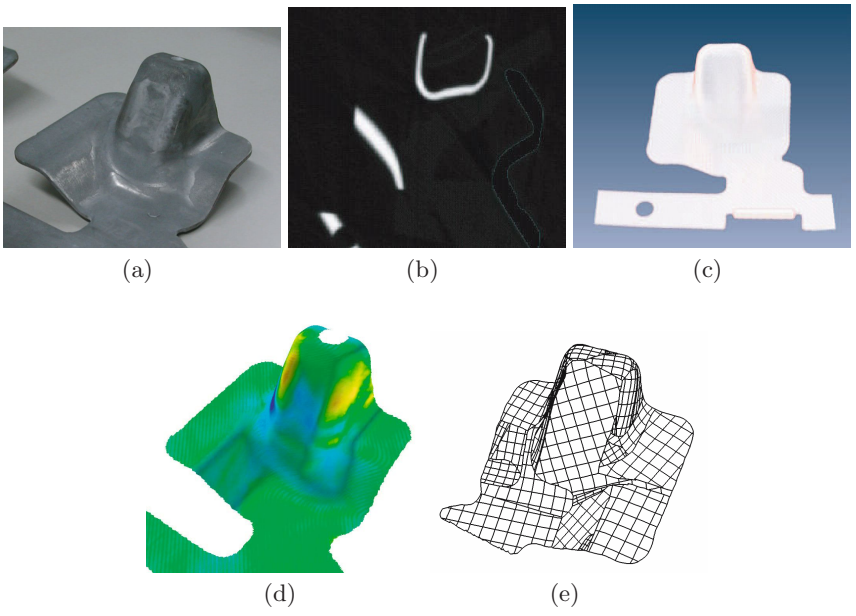


Fig. 1. X-ray CT (Computed Tomography) is used to scan a thin plate structure (a). (b) shows a sectional (slice) image. By building up these sectional images, a volumetric model (c) is constructed. A skeleton surface mesh of this model is generated with thickness distribution (d). (e) shows a parametric surface model generated from the mesh (d).

0.4mm, and the slice thickness is 0.5mm. The size of the volumetric model is typically from 512^3 to 2048^3.

These values are considered less precise than the standard accuracy of the modern optical surface scanners widely used to generate surface point clouds of parts for reverse engineering or inspection. However, CT scanning does provide several advantages. Firstly, it enables scanning of internal structures that cannot be observed from the outside. The sheet metal parts of car bodies often have internal reinforcement structures, and these parts need to be cut to measure the internal portion. This is a very expensive process, and is problematic because it changes the part shape. Secondly, the thickness distribution of sheet metal parts can be measured. This provides useful information for stamping process engineers in particular, who are keen to check for areas of reduced thickness that can cause breakage in the stamping process. The thickness of sheet metal can be measured point by point using traditional instruments, but its distribution cannot be obtained without a CT scanner.

Such scanned images have therefore been widely used in the inspection of mechanical parts. However, recent severe industrial competition for better product quality and a shorter product development time has resulted in a strong demand for the extensive use of such scanned data in *digital engineering systems* such as

CAD (computer-aided design), CAM (computer-aided manufacture) and CAE (computer-aided engineering) [1]. As volumetric models are not convenient for use in these systems, it is necessary to first generate triangular meshes from them to be fed to various digital engineering systems. Fig. 1(e) shows a parametric surface model generated by a reverse engineering system from the mesh in Fig. 1(d).

In this research, we study methods to generate skeleton meshes (i.e. medial surface meshes) from a volumetric model of a thin plate structure [2,3,4] (Fig. 1(d)). Skeleton surfaces are a basic representation of a thin plate structure for engineers, enabling visualization of the thin plate structure with its skeleton surface and thickness distribution. Standard isosurfacing algorithms such as Marching Cubes [5] that are normally used to generate mesh models from volumetric models cannot be applied here, as the skeleton surface is neither an isosurface nor a boundary surface.

In industrial applications, the mesh quality must be high enough for use in the systems involved. First of all, the geometrical accuracy must guarantee good estimates for the error bound. The typical thickness of sheet metal in a car body structure ranges from 0.6mm to several millimeters. However, the voxel size in our experiments is about 0.4mm, so we need a sub-voxel level of precision. In addition, the mesh should have good connectivity between vertices and be free of small holes and bumps, otherwise the mesh must be manually adapted for use in CAD systems.

In our prior studies [2,3] we proposed our basic ideas for extracting medial surfaces [2] and its extension [3]. We also introduced an extended Marching Cubes method for contouring non-manifold mesh [4]. This paper improves these ideas and methods to be more general and consistent using a concept of point labeling.

This paper is organized as follows: Section 2 introduces a brief survey of related works. In Section 3, we introduce basic concepts and methods for generating skeleton surfaces. In Section 4, experimental results obtained by our prototype software are shown for real mechanical parts. Section 5 summarizes the work.

2 Related Work

Our issue here is related to skeletonization, the process of extracting skeleton representations of objects. This area has been extensively studied in various research fields such as computational geometry and image processing.

2.1 Mesh Based Approach

The Medial Axis Transform (MAT) generates a skeleton for an object known as a medial axis. The medial axis for a three-dimensional object is defined as the locus of centers of maximal inscribed spheres [6]. This medial axis for a three-dimensional object is a set of sheets that are connected together. Our skeleton surface corresponds to the subset of these sheets running through the center of

(a) (b) (c)

Fig. 2. MAT for a three-dimensional object (a). (b) A boundary mesh for an object. (c) An approximate medial axis of (b).

the object without branch sheets connecting to the object's corners (Fig. 2(a)). In such a branch sheet, the radius of a maximal inscribed sphere decreases as it approaches the corner and finally vanishes. We can therefore prune them by limiting the radius to be larger than a particular threshold value. In our case for thin plates, the limit is about the half of the thickness of the plates.

The medial axis can be defined and computed in both continuous and discrete domains, i.e. solid objects with smooth boundary surfaces and their discretized representation with binary voxels respectively. Since a boundary mesh of an object can easily be generated using isosurfacing methods [8] from a volumetric model of it obtained by CT scanning, it must be a straightforward way to compute a skeleton surface as a part of the medial axis of an object's boundary mesh.

As one of the most popular topics in computational geometry, the computation of a medial axis has been studied by many researchers. An exact medial axis of a triangular mesh can be computed [9]. However, an exact computation method is expensive for our boundary meshes that have several hundred thousand or even millions of triangles. Most other approaches have been made to compute an approximate medial axis. For polyhedral objects, Etzion and Rappoport [10] proposed a method using an octree subdivision of space. The medial axis can be computed from a Voronoi diagram [11] or by Delaunay triangulation [12]. Many researchers therefore use a set of sample points on the boundary to generate an approximate medial axis with the Voronoi diagram of these points [13,14,15,16,17,18,19,20].

Fig. 2(b) shows a boundary mesh generated using the isosurfacing method, while Fig. 2(c) is a medial axis generated using a method proposed by Yoshizawa [7]. We find that the medial axis is so sensitive to the small undulation of the boundary surface that undesired small bumps in the medial surface are generated as shown in Fig. 2(c), and it is also necessary to exclude branch sheets.

Another serious problem is caused by the limitations of CT scanners for isosurfacing. The images of such scanners have various artefacts [21]. In particular, it can be shown that a given CT image is equivalent to the convolution of an object's ideal representation with a smooth, two-dimensional Gaussian-like function called the point spread function. A graph of CT values (density) across an

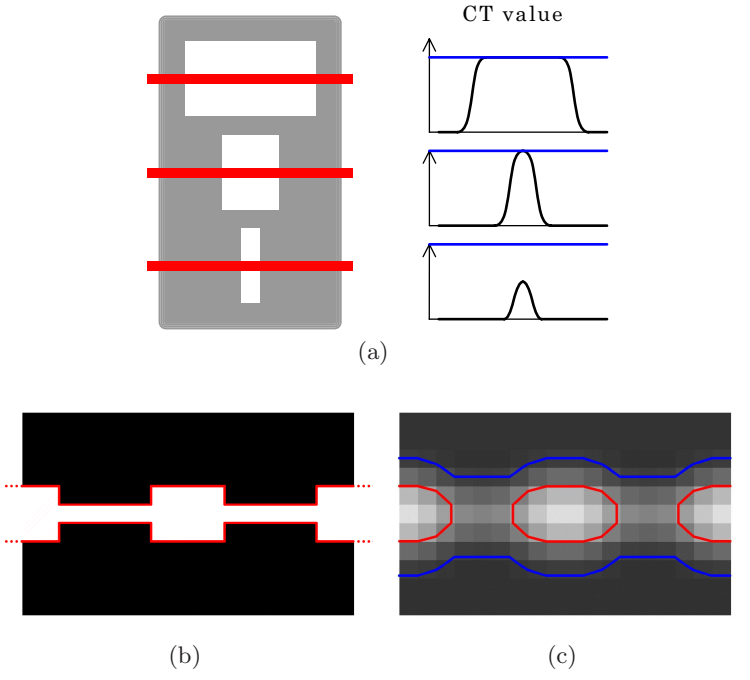


Fig. 3. (a) Relationship between plate thickness and CT value. (b) A section of a thin plate. (c) Isosurfaces generated from a volumetric image of (b).

edge border of the object therefore proceeds in an S-curve shape, gradually increasing from a background value through intermediate values (i.e. unsaturated values) to the full CT value (i.e. the saturated value) of the object's material (see Fig. 3(a)).

In other words, it takes several voxels to increase from the background value to the saturated object value. This does not pose a problem in general for isosurfacing, as isosurfaces can be extracted by setting a threshold value θ_M as the average of the background CT value and the saturated object CT value. However, in the case of a thin plate that has a thickness of only a few voxels, the CT value cannot be saturated even in the interior of the plate. Under these conditions, isosurfacing with averaged threshold values generates unwanted through-holes in the mesh as shown in Figs. 3(b) and (c). Such holes can be avoided by decreasing the threshold value, but the thickness of the object increases.

2.2 A Voxel-Based Approach

A large part of the work for skeletonization is undertaken for binary voxel objects. First, we binarize the volumetric model from CT data using the threshold θ_M. A voxel skeleton can then be extracted in two main ways: a distance-map-based approach and a topological thinning-based approach, as introduced below.

Once a voxel skeleton (a set of binary voxels) is created, a mesh can be obtained by generating triangles directly to each of these skeleton voxels using *cuberille* methods or other polygonization methods developed in the field of volume visualization [22]. A detailed discussion is given in [23].

However, our research concludes that such meshes are of limited use since their level of detail cannot be smaller than the voxel size. Our method, though, is closely related to voxel skeletonization, so a brief survey is given below.

Thinning Algorithms. Sequential topological thinning is performed by successively eliminating the outermost voxels of an object while preserving its topology [24,25,26]. When no more voxels can be eliminated, the algorithm terminates to produce a skeleton.

It is important to prevent some voxels from being eliminated during thinning, otherwise the skeleton is shrunk too much and a very small object results. To extract a skeleton surface, surface-end voxels that represent the end voxels of a skeleton surface must be detected and kept during thinning. These tests can be thought of as trying to locally detect a global feature such as the skeleton surface. As a result, they tend to generate very bumpy objects, though skeletons created by thinning do not have branch sheets connecting to the corners of the object. Tao et al. [27] introduced conditions for end voxels, but the issue is not completely resolved. Zhao [28] introduced a template-based method to control the thinning process, but this can be applied only to two-dimensional problems.

On the other hand, parallel thinning repeats simultaneous elimination of voxel sets, but does not eliminate voxel by voxel as in the case of sequential thinning. It first finds all boundary voxels that can be eliminated, and then deletes them simultaneously. The method can generate objects that are less bumpy, but cannot preserve the topology of the object. Bertrand [29,30] introduced formal conditions for generating a surface skeleton. Tsao et al. [31] introduced a condition of plane-checking for generating a smoother skeleton surface. None of these methods, however, is able to preserve the topology. There are some parallel thinning methods that can preserve topology [32], but unfortunately they use a weak connectivity condition (26-neighbourhood) for the object and produce a bumpy skeleton surface.

Distance-based Algorithms. A distance map is made by computing the distance from each voxel of the object to the nearest boundary voxel of the object. This process is known as distance transform, and has been studied extensively. Excellent surveys can be found in the work of [33] and [34]. In this map, singular voxels determine a medial axis. However, it is not easy for distance based algorithms to preserve the object's topology, and post-processing must be applied to restore the topology [35,36].

Geodesic-Based. Our algorithm is based on the concept of geodesic distance. Costa [37] introduced the idea of using the geodesic distance along the surface of the object to determine skeleton voxels. Based on this idea, Prohaska proposed a method to extract skeleton voxels [38]. For each voxel of an object, we can find the nearest object boundary voxels and compute their geodesic distance

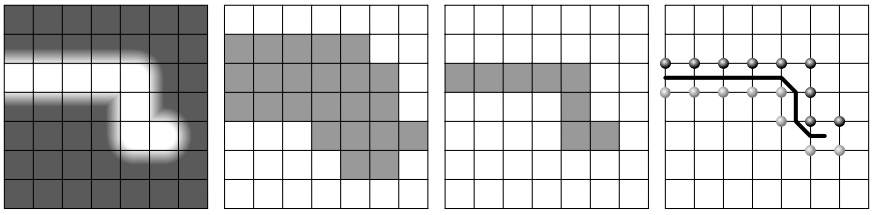


Fig. 4. (a) A volumetric model. (b) Its binary voxel model created by thresholding. (c) Extracted skeleton cells. (d) A skeleton mesh generated by contouring the skeleton cells.

along the boundary of the object. A skeleton voxel is then defined as such if it has neighborhood voxels with geodesic distances larger than a given threshold. From the point of view of MAT, these nearest boundary voxels are considered as tangent points of an inscribing sphere whose center is on the medial axis.

With this approach, the global shape of the object is taken into account to generate a well-shaped surface skeleton, and surface branches reaching toward the corners can be avoided by controlling the threshold value. Day et al. [39] extend this geodesic-based approach to define and extract skeleton curves.

In conclusion, the existing approaches are insufficient for extraction in terms of our purpose of generating skeleton surfaces with sub-voxel accuracy and without branch sheets.

3 Contouring Skeleton Surface

Our input data consists of a volumetric model of an object (Fig. 4(a)) where the voxels are arranged in a regular three-dimensional lattice. We regard a voxel as a lattice point, and define a *cell* as a cube whose eight corners are the lattice points. To achieve sub-voxel accuracy, an approach similar to the Marching Cubes method is taken to generate a triangular mesh (*skeleton mesh*) inside the cells as shown in Fig. 4(d). We therefore first extract *skeleton cells* (in Fig. 4(c)) that a skeleton surface passes through, and use a skeletonization approach for this purpose. Let us discuss how to determine skeleton cells.

3.1 Extracting Skeleton Cells

Let $O \subset R^3$ be an object that is bounded by connected manifold surfaces ∂O. Our target products are thin plate structures that are assumed to take a smooth surface shape and whose thickness also changes very gradually. Fig. 5 shows a sectional image of such a plate-like object O. In this section, we consider only portions of O that lie away from its border edges, and also assume that there are no junctions where more than one plate intersects. This issue will be discussed in Section 3.4.

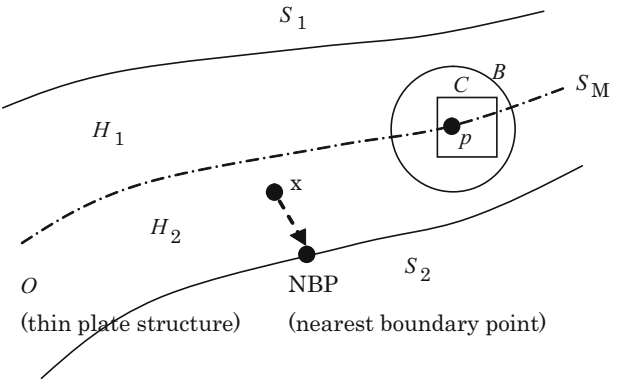


Fig. 5. A section of a thin plate O

For such a portion of a plate-like object, we can define two *side surfaces* S_1 and $S_2(\subset \partial O)$ as shown in Fig. 5 and a skeleton surface $S_M \subset O$. Consider a point $x \in O$ and its nearest points on the boundary ∂O. S_M is defined as a set of points of O that have more than one distinct nearest boundary points (NBP). We rather introduce a function nbp(x) to define one NBP uniquely for each x, including those of S_M (the reason for this will be discussed later). We select the NBP with the smallest coordinate in the order of x, y and z when x has more than one NBP. By using function nbp, the region of O can be divided into two regions, H_1 and H_2, as follows:

$$H_i = \{x \in O | \text{nbp}(x) \ \ on \ \ S_i\}, \ \ \text{for} \ \ i = 1, 2$$

H_1 is an upper-half region of O with boundary S_1, while H_2 is a lower-half region with boundary S_2 in the figure. We define a labeling function for $x \in O$ as $\chi : O \to \{1, 2\}$ by $\chi(x) = i$ if x is a point of H_i. Note that S_M can be non-orientable. Imagine a Moebius band with some thickness; the two boundary surfaces cannot be globally distinguished, but they are only required to be identified locally.

Consider a point $p \in S_M$ and its open-ball neighborhood $B(p)$ of radius R. The intersection of S_M and B is homeomorphic to a disk for some value of R. We assume $B(p)$ is completely involved in O, or in a practical sense, the thickness of the plate is bigger than the sampling interval of the volumetric model. S_M then divides B into two distinct parts, each of which is a subset of either H_1 or H_2.

Within a neighborhood of $B(p)$, we locate a cubic cell C of eight corner points $p_1, p_2, \ldots, p_8$, and compute their labels $\chi(p_i)$. When C intersects with S_M, $\chi(p_i)$s take labels of both 1 and 2. Such a cell is known as a *skeleton cell*.

All the possible labeling patterns for a skeleton cell are shown in Fig. 6. In this figure, each pattern is a representative of those that are symmetrically equivalent to it. As S_M is located between H_1 and H_2, it passes between two points with different labels (1 and 2). The patterns in Fig. 6 are part of those of

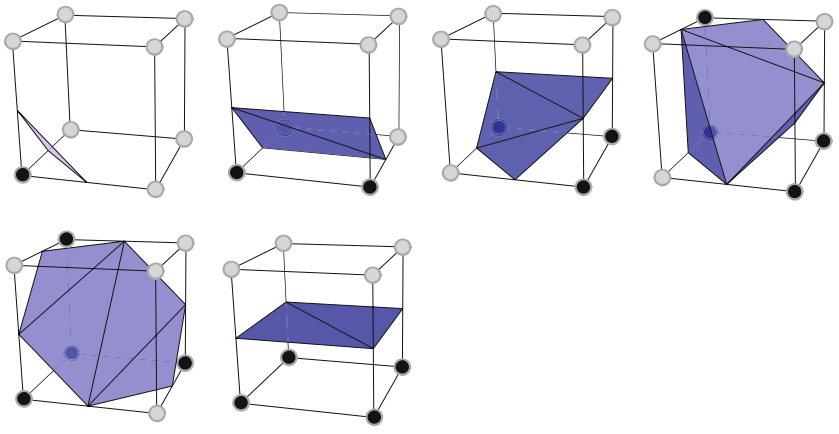


Fig. 6. Possible labeling patterns for a skeleton cell

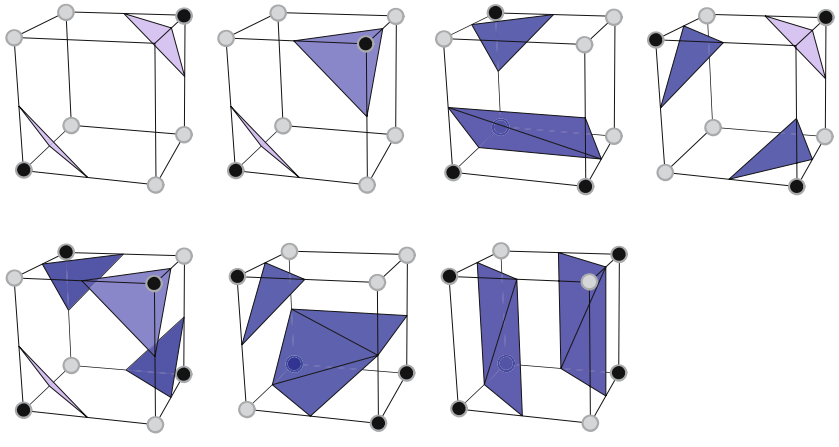


Fig. 7. Impossible labeling patterns for a skeleton cell

the Marching Cubes algorithm. The other patterns in Fig. 7 are not allowed, as these configurations require that S_M passes the cell twice or more times, which contradicts our assumption. In the patterns in Fig. 6, each set of points with the same label is either a single-point set or a 6-connected component where a point is connected by an edge to some other point of the component.

By regarding the different labels 1 and 2 as plus and minus signs, a triangle mesh can be generated in cell C by interpolating the mid-points of intersecting edges. The patterns in Fig. 7 include well-known "ambiguous" patterns in the Marching Cubes [40] method, but can be neglected here as invalid for S_M.

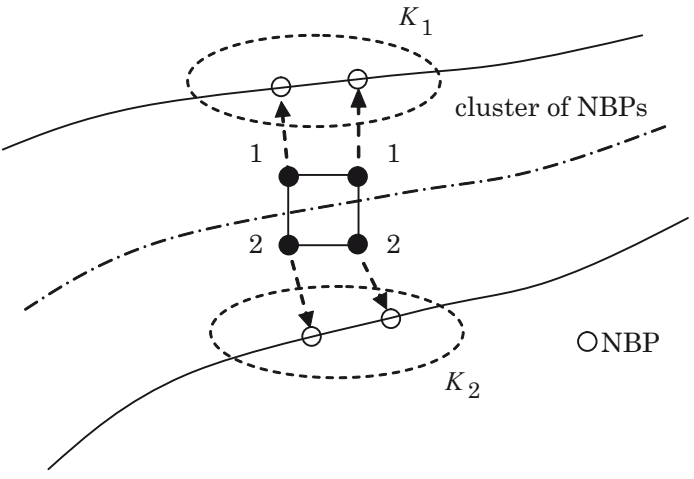


Fig. 8. Clustering NBPs for labeling

3.2 Point Labeling χ

We now discuss how to define χ, which is dependent on distinct side surfaces S_1 and S_2 that cannot be defined globally as outlined above. Instead, we use the geodesic distance between two points of ∂O, which is the length of the shortest path on the ∂O between these two points.

To define χ for each of the corner points p_i of a cell C, we first compute a set of NBPs $K = \{\text{nbp}(p_i)\}$ for all p_i of C (Fig. 8). We then cluster K into disjoint sets $K_j, (j = 1, 2)$ so that for each pair of points in K_i the geodesic distance is less than the given threshold θ_{GD}. By this clustering, NBPs on the same side of the object can be collected into a cluster, because the geodesic distance between boundary points on the same side is small, while that between those on different sides is large. $\chi(p_i)$ is defined on the basis of which K_j p_i belongs to.

As labels are defined locally to a cell, it is problematic that the orientation of S_M can be inconsistent from cell to cell even though S_M is orientable. It is possible to globally cluster all the NBPs to globally determine consistent labels for an orientable S_M, but this is computationally expensive. In practice, a procedure is used to make triangle orientation consistent by simply propagating the orientation from a triangle to the whole mesh by flipping normal vectors of the triangles if necessary.

3.3 Contouring Skeleton Meshes

Using these labels, the Marching Cubes algorithm is applied with trivial mapping to insert a vertex in the middle of the edge (Fig. 9(a)). However, as the surface obtained by the trivial mapping exhibits serious aliasing, we also apply interpolating mapping to insert a vertex v on edge $< p_i, p_j >$ so that the two

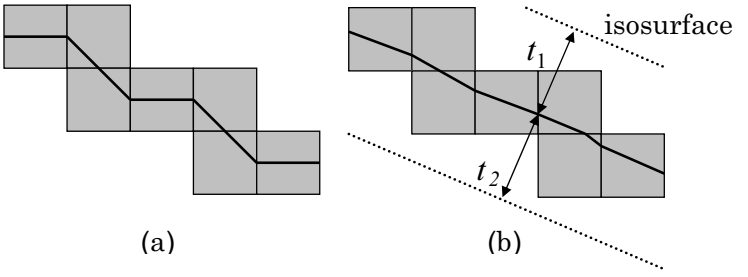


Fig. 9. Applying the Marching Cubes method: (a) is a two-dimensional example of trivial mapping, while (b) shows an example of interpolating mapping

distances t_0 and t_1 from the vertex to the boundary mesh of the plate structure are equal (Fig. 9(b)). As the first approximation, the distances to nbp(p_i) and nbp(p_j) can be used if p_i and p_j have different labels. For more accurate interpolation, a boundary mesh is generated by isosurfacing the volumetric model of the plate structure. As outlined in Section 2, when the plate structure is very thin, the boundary surface may contain holes because of the unsaturated CT value. In such cases, the threshold value is lowered to fill the holes. We then search for tangent points from v to the surface mesh, limiting the search to cells that include nbp(p_i) and nbp(p_j). In both cases, the interpolated position of v is limited to be on the edge $< p_i, p_j >$.

3.4 Junctions of Plates

Extending the above labeling function to cases where N plates meet as shown in Fig. 10 is straightforward. In these cases, labeling function χ is expected to take N labels from 1 to N, and such a skeleton cell C will contain points from more than two different labels. The condition on the connectivity of the points of a cell mentioned above also holds for such cases.

The Marching Cubes method and other isosurface contouring methods are only applicable to cells with two different labels (signs) to generate two manifold surfaces. In a skeleton cell with more than two labels, more than two skeleton surfaces intersect, making the skeleton surface non-manifold. A *non-manifold contouring method* is needed for such non-manifold skeleton cells. The Generalized Marching Cubes method [41] derives non-manifold meshes by further dividing the cell into a regular lattice of smaller cells to define meshing patterns. However, this is a very expensive process, and there is no clear reason why these patterns do not have ambiguity. Dual Contouring [42] is based on the idea of generating triangles around edges by connecting the edge point to points inside its four incident cells. This is an excellent method, in that it can handle not only regular lattices of cells but also adaptively generated,

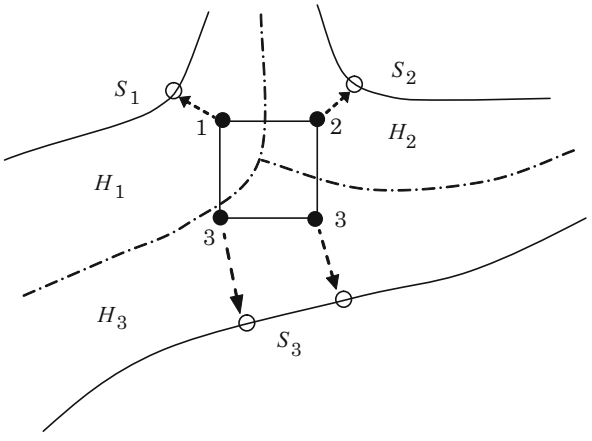


Fig. 10. Labeling in a junction area where more than two plates meet

non-uniform lattices such as octrees. However, in order to define the proper positions of the cell points, normal vectors (Hermite data) at edge points must be estimated. In our applications, such Hermite data cannot be easily estimated from CT data.

We extend the meshing table of the Marching Cubes method (shown in Figs. 6 and 7 to include these non-manifold cases, and use Eberly's algorithm [43] to derive meshing patterns for non-manifold cells as follows. In Fig. 11, graphs are defined on the six faces of a cell. A vertex (an edge vertex) is inserted on an edge of a cell connecting two vertices with different labels. If there are two edge vertices on a face, they are connected. If there are more than two edge vertices on a face, a new vertex (a face vertex) is inserted in the center of the face, and edge vertices on the edges are connected to this face vertex. These form a graph with a "T" shape or a "+" sign on the face. The graphs on the six faces constitute an *edge mesh*, which is then triangulated by *ear clipping* [44] to generate a triangular meshing pattern for the cell. Ear clipping is a simple triangulation method that repeats clipping of a triangle from the edge mesh as shown in Fig. 11. At every iteration, a vertex sharing only two edges (valence 2) is chosen, and a triangle is defined by this vertex and the two edges. A third edge is inserted if necessary. The triangle is then clipped by deleting this vertex and the two sharing edges, and the process is repeated until there are no more vertices of valence 2. The collection of clipped triangles can be used for contouring the cell (Fig. 11).

This method is used to pre-compute a meshing table similar to that of the Marching Cubes method. We first enumerate all the configurations of n kinds of label on eight corner points of a cell. For instance, Figs. 6 and 7 represent distinct patterns of configurations of two labels. These patterns are chosen

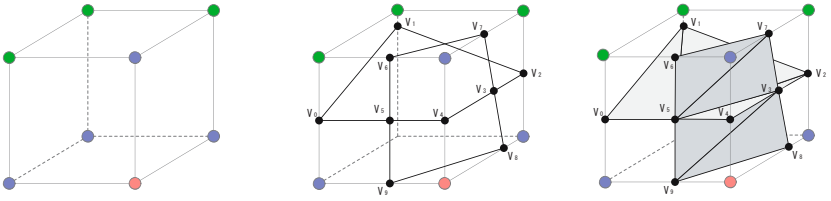


Fig. 11. Generating a meshing pattern. First, an edge mesh (middle) is defined for a cell with labeled vertices (left). Ear clipping defines a set of triangles (right).

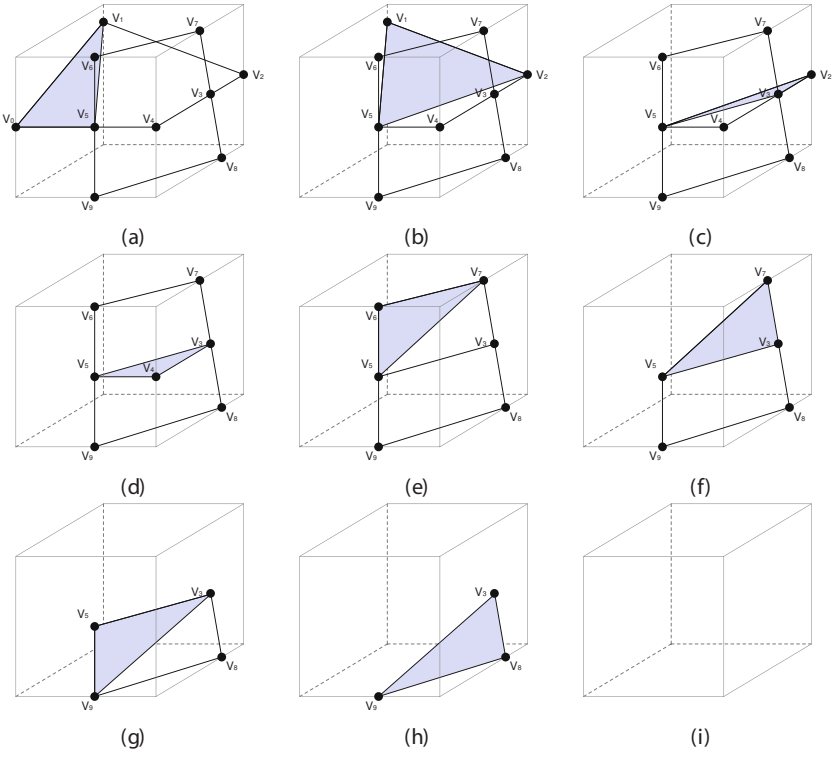


Fig. 12. Ear clipping procedure for triangulating an edge mesh in Fig. 11

from all the possible 2^8 patterns by excluding those that are symmetrically equivalent. These 14 patterns are complete for $n = 1$ (single label) and $n = 2$ (two labels).

All the distinct patterns for $n = 3, 4, \ldots, 8$ are enumerated, and Table 3.4 shows the numbers of such patterns. For instance, case $n = 3$ has 44 distinct patterns in total consisting of 5 different subcases. In the first subcase, $(1, 1, 6)$ means that the numbers of points of the three different labels are 1, 1 and 6.

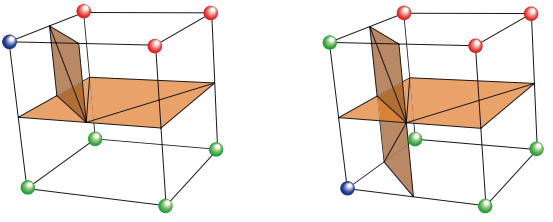


Fig. 13. Some examples of non-manifold meshing patterns for $n = 3$ with $(1, 3, 4)$

Table 1. The numbers of distinct patterns for labeling points of a cell with n labels

n						Total
1	(0,0,8) 1					1
2	(0,1,7) 1	(0,2,6) 3	(0,3,5) 3	(0,4,4) 6		13
3	(1,1,6) 3	(1,2,5) 6	(1,3,4) 10	(2,2,4) 12	(2,3,3) 13	44
4	(1,1,1,5) 3	(1,1,2,4) 16	(1,1,3,3) 13	(1,2,2,3) 22	(2,2,2,2) 12	66
5	(1,1,1,1,4) 6	(1,1,1,2,3) 17	(1,1,2,2,2) 20			43
6	(1,1,1,1,1,3) 3	(1,1,1,1,2,2) 12				15
7	(1,1,1,1,1,1,2) 3					3
8	(1,1,1,1,1,1,1,1) 1					1
Total						186

And this subcase has 3 different patterns. For the current implementation, only the case of $n = 3$ is programmed to define meshing patterns. Figure 13 shows some examples of meshing patterns for $n = 3$ with $(1, 3, 4)$.

Eberly's method was extended in line with our previous experience in implementation. First, the algorithm generates a triangle on a cell face as shown in Fig. 14. Such triangles are not appropriate for a skeleton mesh, and can be avoided by first selecting vertices that are incident to a face vertex. Second, the ear-clipping process terminates when there is no vertex of valence 2 in the edge mesh, even if some vertices and edges remain, as shown in Fig. 15. We have not yet succeeded in analyzing the issue of a condition to detect such cases and a method to clip all the vertices. We propose a method whereby a vertex is inserted in the center of the cell and connected to the face vertices of the edge mesh as shown in Fig. 15.

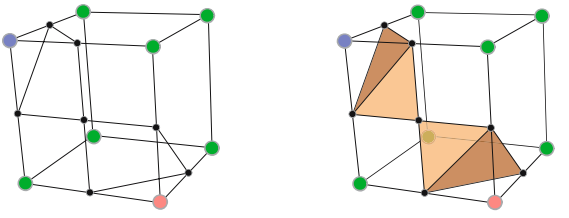


Fig. 14. Problem of the ear-clipping process: triangles are defined on faces

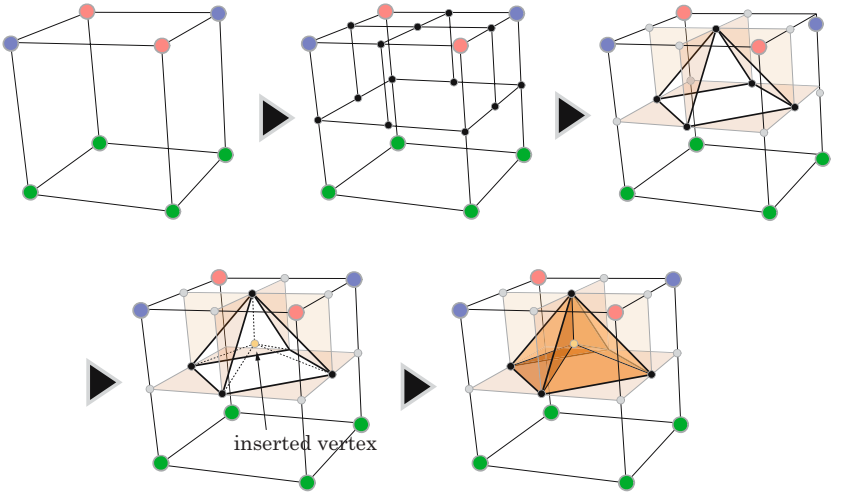


Fig. 15. Problem of the ear-clipping process: clipping process halts before all the edge mesh is eliminated (upper right figure). By inserting a vertex in the center of the cell and connecting it to the face vertices of the edge mesh, the process can be continued.

3.5 Characteristics of Skeleton Cells

Let us consider the whole set of skeleton cells C_M extracted in the previous section. The skeleton cells are regarded as foreground, and the others as background. Away from the boundaries of S_M, we observe that C_M satisfies the following two conditions:

1. C_M is tunnel free.
2. every cell of C_M that is not an edge cell is a surface cell.

See the Appendix for definitions of the concepts used above. The matter can be intuitively explained as follows. The first condition is based on the assumption that S_M does not contain a hole, since it is a collection of two-manifold surfaces with boundaries. The second condition is supported by the assumption that each part of S_M separates the object into two halves. A set of cells that satisfies these conditions is defined as a *cubic shell*.

According to observation and experimentation, our method can generate C_M to satisfy the cubic shell conditions almost everywhere, even for noisy CT data. This is due to the labeling function χ that can uniquely determine the label of a point, and thus does not allow a skeleton surface tangent to the corner points or edges of a cell. When some part of C_M does not satisfy the cubic shell conditions, C_M can be adapted by adding and removing skeleton cells [2].

4 Implementation and Experimental Results

4.1 Implementation Techniques

NBPs and the geodesic distance are computed by approximating an object O as a binary voxel model. What we are given is a volumetric model of a thin-plate object (Fig. 4(a)), and this is converted by thresholding into a binary voxel model on which NBP and geodesic distances are computed (Fig. 4(b)). For distance computation, a (3, 4, 6) weighted metric is used to approximate the Euclidean distance [25].

As the current algorithm to determine a skeleton cell outlined in the previous section is expensive to compute, we propose a cheaper method based on the fact that a skeleton cell is one of the patterns shown in Fig. 6. Consider four diagonal pairs of cell corner points. In these patterns, at least one diagonal pair has different labels. Therefore, if the distance between the NBPs of the points of a cell's diagonal pair is lager than θ_{GD}, the cell can be a skeleton cell [2]. We thus check these four diagonal pairs; this is a necessary condition, but invalid patterns can be rejected by the labeling process that follows.

By relying on the cubic shell conditions, point labels can be determined without using the NBP clustering proposed in the previous section. We instead analyze the connected components of background cells around a skeleton cell by *neighborhood polarity analysis* as proposed in [2]. Since a skeleton cell is a surface cell, its neighborhood has two or more 6-connected components. Cell points are labeled according to the components in which they are involved. These two techniques accelerate the computation.

We now discuss how to determine θ_{GD}. As outlined in Section 2, if it is too small, the skeleton surface contains branch sheets around the edge of the plate. Based on the results of experimentation, we use

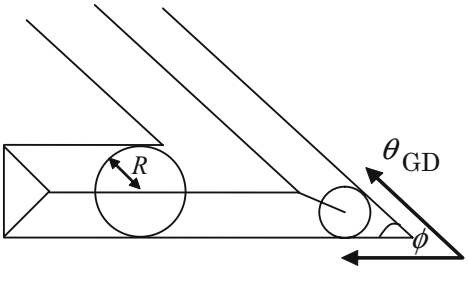


Fig. 16. Estimation of θ_{GD} to avoid branch sheets

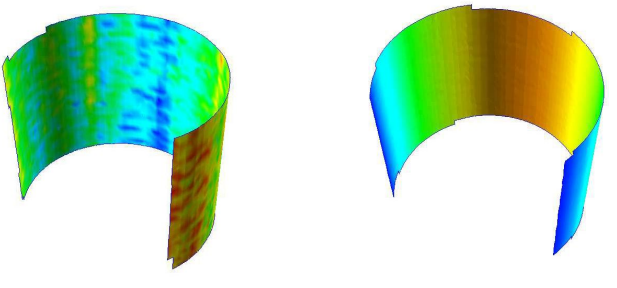
$$\theta_{\mathrm{GD}} = \frac{R}{\arctan(\phi/2)} + 2\Delta$$

considering the cases shown in Fig. 16 where two plates form a sharp angle. R is the lower bound of the radius of the inscribed spheres of the medial axis, thus avoiding the branch sheets mentioned in Section 2, and ϕ is the angle of a corner of the plate to be extracted. Δ is the maximum of the three dimensions of the voxel, and a value of 2Δ is added to relax the binarization error. R and ϕ are user-defined.

4.2 Experimentation

Prototype systems were implemented on a standard PC equipped with a 2.4GHz Pentium 4 and 1GB of memory, and various test pieces were made to evaluate the accuracy of our method. Fig. 17 shows two kinds of cylindrical test piece. Test-piece A has a uniform thickness, while the thickness of B is variable. Fig. 18 shows the extracted skeleton surfaces, with the applied color representing the thickness

Fig. 17. Test pieces A (left) and B (right) for evaluating the accuracy of skeleton surfaces

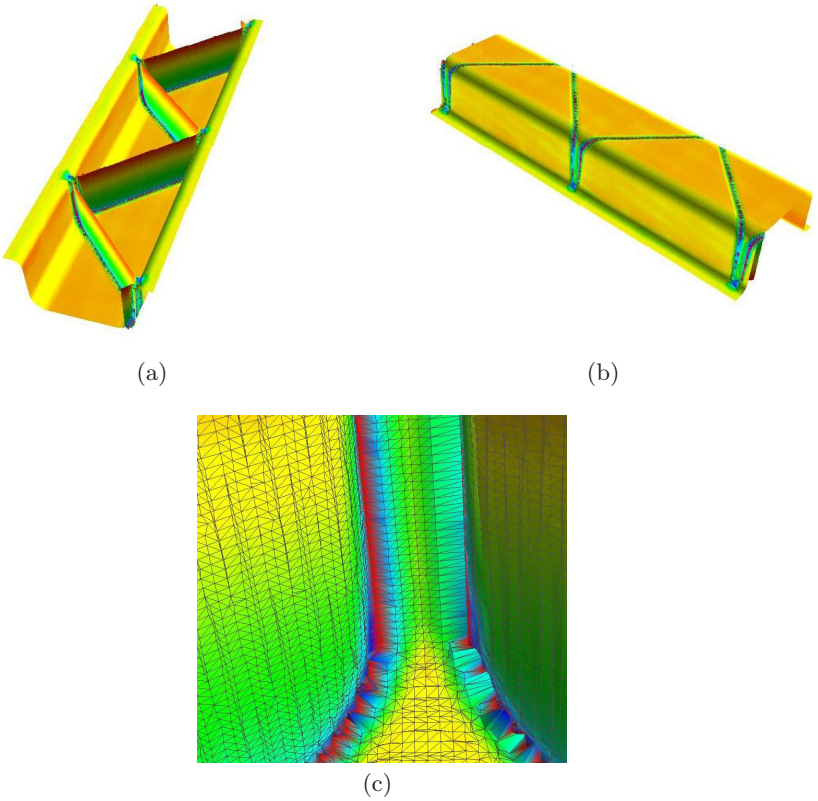


(a) (b)

Fig. 18. Skeleton meshes for the test pieces with thickness distribution

(a) (b) (c)

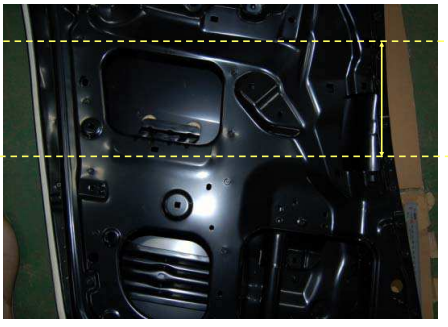
Fig. 19. (a) is a photograph of the target aluminum pipe. (b) and (c) are generated skeleton surfaces with thickness distribution raiging from 0mm to 5mm.



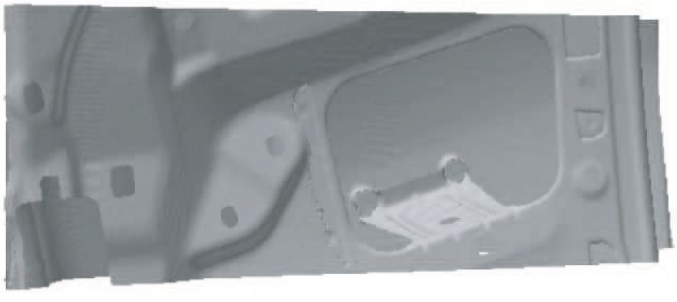
(a) (b)

(c)

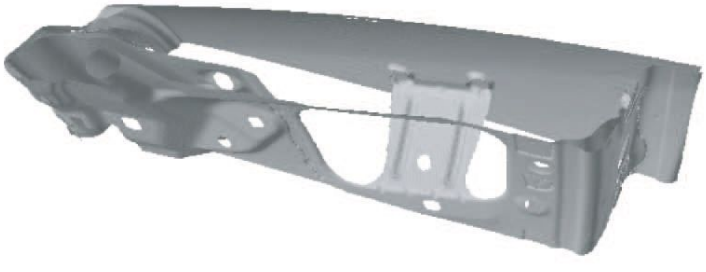
Fig. 20. Skeleton meshes (a), (b) for a thin plate structure with T-junctions. (c) Close-up of the non-manifold part of the mesh.



(a)



(b)



(c)

Fig. 21. Photograph of a door panel. (a) The designated portion is scanned. (b) and (c) show the skeleton meshes of about 2.9 million triangles.

distribution. The thickness is computed using the surface mesh discussed in Section 3.3. Test-piece A has a rather uniform distribution, while that of B is smoothly variable. The voxel size is 0.4mm × 0.4mm × 0.5mm, and the volume model has 100 × 100 × 50 voxels. The surface mesh consists of about 12,000 triangles in both cases. We also evaluated geometrical tolerance values. The radius of A is 8.47mm (with a true value of 8.5mm), and its cylindricity

is 0.075mm. These values were evaluated using a commercial Computer Aided Testing (CAT) software. These results show that our method is able to generate a sufficiently accurate skeleton mesh with respect to the voxel sizes of the volume model.

Fig. 19(a) shows a part made by deforming an aluminum pipe. The heavy deformation process changes the thickness of the material, as can be easily understood from our skeleton surfaces with thickness distribution (b) and (c). It took 100 seconds to extract a skeleton surface for CT data of $290 \times 90 \times 270$ voxels. Again, the thickness distribution is also computed.

Fig. 20 shows another example with T-junctions. The skeleton surface thus becomes a non-manifold mesh. The size of the volume model is $200 \times 200 \times 288$, with a voxel size of $0.626\text{mm} \times 0.625\text{mm} \times 1.0\text{mm}$. It takes about 13 minutes to generate 257,571 triangles.

Finally, Fig. 21 shows a complex example of a car door panel. The thickness of the panel is about 0.8mm, and the voxel size is $0.4\text{mm} \times 0.4\text{mm} \times 0.5\text{mm}$. It means that the panel has thickness of only two voxels. By lowering the threshold value for binarizing the volumetric model, our method succeeded in extracting a skeleton mesh, though the plate is too thin for the thickness to be computed. The number of triangles is about 2.9 million. It should be noted that this panel is also non-manifold because the plates are welded together to make junctions.

5 Conclusions

This report has presented a method to contour a skeleton surface of triangular mesh from a volumetric model of a thin plate structure scanned using X-ray CT. A geodesic-based skeletonization algorithm was used to select skeleton cells. We also introduced a function to determine labels for object points, which is used to determine the labels of a cell's corner points. The skeleton cells are then contoured using a method that extends the Marching Cubes algorithm to enable handling of multiple labels (signs) to generate non-manifold meshes. The generated mesh is optimized to pass through the mid of the object by interpolating mapping. Some experimental results for industrial samples were also shown using our implemented prototype system, and it was proven that the accuracy of the generated mesh is sufficient for industrial applications.

Acknowledgment

I would like to thank my former students, Yasuhiro Tamaki and Yohei Kobayashi, for their valuable contribution to this research work. This research was in part supported by Toyota Motor Corp. and Hitachi Ltd. and also in part by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science, No. 15360080.

References

1. Suzuki, H.: Convergence engineering based on X-ray CT scanning technologies. In: Proceedings of JSME Digital Engineering Workshop, pp. 74–77 (2005)
2. Fujimori, T., Suzuki, H., Kobayashi, Y., Kase, K.: Contouring medial surface of thin plate structure using local marching cubes. Journal of Computing and Information Science in Engineering (Short Paper) 5(2), 111–115 (2005)
3. Fujimori, T., Suzuki, H.: Medial surface extraction and extension from CT data. In: Proceedings of the 13th Pacific Conference on Computer Graphics and Applications (short paper) (2005)
4. Fujimori, T., Suzuki, H.: Surface extraction from multi-material CT data. In: Proceedings of CAD/Graphics 2005 (2005)
5. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp. 163–169. ACM Press, New York (1987)
6. O' Rourke, J.: Computational geometry in C, 2nd edn. Cambridge University Press, Cambridge (2001)
7. Yoshizawa, S., Belyaev, A.G., Seidel, H.P.: Free-form skeleton-driven mesh deformations. In: SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications, pp. 247–253 (2003)
8. Bloomenthal, J.: Polygonization of implicit surfaces. Computer Aided Geometric Design 5(4), 341–355 (1988)
9. Culver, T., Keyser, J., Manocha, D.: Exact computation of the medial axis of a polyhedron. Computer Aided Geometric Design 21(1), 65–98 (2004)
10. Etzion, M., Rappoport, A.: Computing Voronoi skeletons of a 3D polyhedron by space subdivision. Technical report (1999)
11. Aurenhammer, F.: Voronoi diagrams - a survey of a fundamental geometric datastructure. ACM Computing Surveys 23(3), 345–405 (1991)
12. Reddy, J.M., Turkiyyah, G.M.: Computation of 3D skeletons using a generalized Delaunay triangulation technique. Computer Aided Design 27(9), 677–694 (1995)
13. Amenta, N., Choi, S., Kolluri, R.: The power crust. In: Proceedings of Solid Modeling '01, pp. 249–260 (2001)
14. Attali, D., Lachaud, J.O.: Delaunay conforming iso-surface, skeleton extraction and noise removal. Comput Geom: Theory Appl 19(2–3), 175–189 (2001)
15. Attali, D., Montanvert, A.: Computing and simplifying 2D and 3D continuous skeletons. In: Comput Vision Image Understanding, pp. 261–273 (1997)
16. Brandt, J., Algazi, V.: Continuous skeleton computation by Voronoi diagram. Comput Vision Graphics Image Process 55, 329–338 (1992)
17. Sheehy, D., Armstrong, C., Robinson, D.: Shape description by medial axis construction. IEEE Trans. Vis. Comput. Graph. 2(1), 62–72 (1996)
18. Turkiyyah, G., Storti, D., Ganter, M., Chen, H., Vimawala, M.: Contouring medial surface of thin plate structure using local marching cubes. Computer Aided Design 29(1), 5–19 (1997)
19. Dey, T., Zhao, W.: Approximate medial axis as a Voronoi subcomplex. In: 7th ACM Symposium on Solid Modeling Applications, pp. 356–366. ACM Press, New York (2002)
20. Dey, T., Zhao, W.: Approximating the medial axis from the Voronoi diagram with a convergence guarantee. Algorithmica 38(1), 179–200 (2003)
21. ASTM: E 1441-00 Standard Guide for Computed Tomography (CT) Imaging. ASTM (2000)

22. Elvins, T.T.: ACM SIGGRAPH 94, Course Notes, Introduction to Volume Visualization: Imaging Multi-dimensional Scientific Data (1994)
23. Lachaud, J.O., Montanvert, A.: Continuous analogs of digital boundaries: A topological approach to iso-surfaces. Graphical Models and Image Processing 62, 129–164 (2000)
24. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing 48(3), 357–393 (1989)
25. Lohmann, G.: Volumetric Image Analysis. John Wiley & Sons, Ltd., West Sussex (1998)
26. Lee, T., Kashyap, R., Chu, C.: Building skeleton models via 3-D medial surface axis thinning algorithms. CVGIP: Graphical Models and Image Processing 56(6), 462–478 (1994)
27. Ju, T., Baker, M.L., Chiu, W.: Computing a family of skeletons of volumetric models for shape description. Proceedings of Geometric Modeling and Processing 20, 235–247 (2006)
28. Dougherty, E.R., Lotufo, R.A.: Hands-on Morphological Image Processing. SPIE Press (2003)
29. Bertrand, G.: A parallel thinning algorithm for medial surfaces. Pattern Recogn. Lett. 16(9), 979–986 (1995)
30. Lohou, C., Bertrand, G.: A new 3D 6-subiteration thinning algorithm based on p-simple points. In: DGCI, pp. 102–113 (2002)
31. Tsao, Y.F., Fu, K.S.: A parallel thinning algorithm for 3-D pictures. Computer Graphics and Image Processing 17(4), 315–331 (1981)
32. Manzanera, A., Bernard, T.M., Preteux, F.J., Longuet, B.: Medial faces from a concise 3D thinning algorithm. In: ICCV 1, 337–343 (1999)
33. Cuisenaire, O.: Distance Transformations: Fast Algorithms and Applications to Medical Image Processing. PhD thesis, Université cotholique de Louvain, Laboratoire de Telecommunications et Teledetection (1999)
34. Jones, M., Satherley, R.: Shape representation using space filled sub-voxel distance fields. In: International Conference on Shape Modeling and Applications, pp. 316–325 (2001)
35. Gagvani, N.: Paramter-controlled volume thinning. Graphical Models and Image Processing 61(3), 149–164 (1999)
36. Malandain, G., Fernandez-Vidal, S.: Euclidean skeletons. Image and Vision Computing 16(10), 317–327 (1998)
37. da F. Costa, L.: Multidimensional scale space shape analysis. In: IWSNHC3DI'99, Santorini, Greece, pp. 214–217 (1999)
38. Prohaska, S., Hege, H.C.: Fast visualization of plane-like structures in voxel data. In: Proceedings of the conference on Visualization '02, pp. 29–36. IEEE Computer Society, Los Alamitos (2002)
39. Dey, T.K., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. In: Polthier, K., Sheffer, A. (eds.) Eurographics Symposium on Geometry Processing, pp. 143–152 (2006)
40. Nielson, G.M., Hamann, B.: The asymptotic decider: resolving the ambiguity in marching cubes. In: Proceedings of the 2nd conference on Visualization '91, pp. 83–91. IEEE Computer Society Press, Los Alamitos (1991)
41. Hege, H.C., Seebass, M., Stalling, D., Zöckler, M.: A generalized marching cubes algorithm based on non-binary classifications. Technical report, Konrad-Zuse-Zentrum für Informationstechnik Belin (1997)
42. Ju, T., Losasso, F., Schaefer, S., Warren, J.: Dual contouring of Hermite data. ACM Transactions on Graphics 21(3), 339–346 (2002)

43. Eberly, D.: Level set extraction from gridded 2D and 3D data (Magic Software, Inc.) URL: http://www.magic-software.com
44. Eberly, D.: Triangulation by ear clipping (Magic Software, Inc.) URL: http://www.magic-software.com

Appendix: Cell Topology

Some definitions of important concepts of cell topology used in this paper are summarized.

cell. A volumetric model is defined on a three-dimensional regular lattice of voxels or points. A cell is a cube consisting of eight corner points (vertices), twelve edges and six faces. Cells inside an object are called foreground cells, while those in empty space are referred to as background cells.

neighborhoods. A neighborhood of x is generally denoted by $N(x)$, which is a set of $3 \times 3 \times 3$ cells centered at x. $(6, 18, 26)$-neighborhoods can be defined by a set of cells that share a face, an edge and a vertex with x respectively.

connected component. Two cells x, y are said to be n-adjacent if, and only if, they are n-neighbors of one another. An n-path is a sequence of cells $(x_1, \ldots, x_k)$ that are n-adjacent pairwise, i.e. x_i is n-adjacent to x_{i+1} for $i = 0, \ldots, k-1$. A set of cells X is n-connected if for any two cells in X there is an n-path in X between these two cells.

surface cell and boundary cell. If the number of 6-connected components in the background cells of $N(x)$ is two or more, x is a surface cell. Likewise, if the number is 1, x is a *boundary cell*. If the number is 0, c is an *internal cell*.

tunnel-free. Given a background cell b, if there is a background cell b' which is edge-adjacent to b, and b' does not belong to the same connected component in background neighborhood $N(b)$, we say that cell b has a tunnel edge. In this context, the tunnel edge is shared by b and b'. Similarly, if there is a background cell b' which is vertex-adjacent to b, and b' does not belong to the same connected component in background neighborhood $N(b)$ of b, b and b' share the tunnel vertex. If in a set of foreground cells, X is said to be *tunnel-free* when its cells do not contain any tunnel edges or vertices.

Parallel Tangency in $\mathbb{R}^3$

John P. Warder

Department of Mathematical Sciences,
The University of Liverpool, Peach Street, Liverpool L69 7ZL, UK
p.warder@liverpool.ac.uk

Abstract. In this paper we examine the mathematics behind pairs of surface points with parallel tangent planes in $\mathbb{R}^3$. We look at the case of disjoint surface pieces, exploring the maps linking the parameters at the two points of tangency and some of their singularities. We go on to consider the same issues for the local case, considering pairs of points with parallel tangent planes in the neighbourhood of a parabolic point on a single surface piece. For the latter case we also consider the effect of the parabolic point being a cusp of Gauss and go on to describe the arrangement of various special curves on the surface in a neighbourhood of the cusp of Gauss. In the second part of the paper we consider surfaces constructed from the chords joining parallel tangent pairs. Giblin and Zakalukin [3] have investigated the envelope of such chords, but here we consider the "equidistants" which are surfaces formed by points at a fixed proportion along the chords. These affinely invariant surfaces are a type of symmetry construction and as such we pay particular attention to the half way equidistant or Mid-Point Tangent Surface (MPTS). We describe its structure in the disjoint surface pieces and local cases and show how it behaves quite differently from other equidistants.

Keywords: Surface, Parallel Tangents, Symmetry Constructions.

1 Introduction

There are many constructions available to measure the local symmetry of two and three dimensional shapes. Two of the most well documented are the Symmetry Set and Medial Axis which are formed as the centres of circles or spheres bitangent to the shape. Both of these constructions have important applications in computer vision but they are restricted by their dependence on circles and spheres whose essential properties are lost under all but euclidean transformations of $\mathbb{R}^3$. As a result we are motivated to study symmetry constructions which are invariant under the much wider range of affine transformations of $\mathbb{R}^3$. One such construction is the Centre Symmetry Set (CSS) which is formed as the envelope of chords joining parallel tangent pairs on the shape. It measures the departure of the shape from central symmetry and since its construction only depends on parallelism it is affinely invariant. The CSS and its singularities have been extensively studied by Giblin and Zakalukin [3] so here we consider a closely

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 465–479, 2007.
© Springer-Verlag Berlin Heidelberg 2007

related affinely invariant family of constructions called *equidistants*. These are formed by points at a fixed proportion along chords joining parallel tangent pairs and we find that the CSS is swept out by the singularities of this family, much as the evolute or focal set is swept out by the singularities of offsets to curves or surfaces.

The paper is divided into two parts: the first part concentrates on the set of pairs of points on a surface (or surfaces) with parallel tangent planes and the properties of maps linking the parameters describing such points; the second part looks at the equidistants formed using this set. We concentrate on the so called Mid-Point Tangent Surface (MPTS), which is formed as the midpoint of chords joining parallel tangent pairs. We study the structure of the MPTS for (i) disjoint surface pieces with one point of tangency on each piece, and (ii) the local case with both points on the same surface piece in the neighbourhood of a parabolic point. Before describing the initial geometry that we will use in each of these cases we set the scene by recalling some basic differential geometry of surfaces and introducing some important notation.

1.1 Some Basic Notions

Let p be a smooth point on a surface piece M in $\mathbb{R}^3$. If we perform euclidean transformations which take p to the origin and rotate so that the xy-plane is tangent to M here, then M is said to be in *special Monge form*. In this form our calculations are simplified as M can be locally described as the graph of a function $z = f(x, y)$ where $f = f_x = f_y = 0$ and the unit normal $\mathbf{n} = (0, 0, 1)$ at $p = \mathbf{0}$. When f is a smooth function in x and y we can write

$$f(x, y) = (a_0x^2 + a_1xy + a_2y^2) + (b_0x^3 + b_1x^2y + b_2xy^2 + b_3y^3) + \text{h.o.t.} \quad (1)$$

where "h.o.t." is used throughout the paper to designate all terms in a series expansion of higher degree than those displayed before it. The behaviour of M in a neighbourhood of p is generically of three distinct types depending on the intersection of M with its tangent plane here: (i) p is called an *elliptic point* if M meets its tangent plane in an isolated point, e.g. $f(x, y) = x^2 + y^2$ at $p = \mathbf{0}$. (ii) p is called a *hyperbolic point* if M meets its tangent plane in two curves crossing transversely, e.g. $f(x, y) = x^2 - y^2$ at $p = \mathbf{0}$, and (iii) p is called a *parabolic point* if M meets its tangent plane in a cuspidal curve, e.g. $f(x, y) = x^2 + y^3$ at $p = \mathbf{0}$. The upper half of figure 1 shows these three contact types.

If we consider the family of planes through the origin which contain the z-axis then each will intersect M forming a curve in that plane. The signed curvature of such curves at $\mathbf{0}$ are called *sectional curvatures* of M at $\mathbf{0}$. Their extrema, say κ_1 and κ_2, are called *principal curvatures* and the directions in which they occur are called *principal directions*. Also, the product $K := \kappa_1\,\kappa_2$ is called the *Gauss curvature* of M at $\mathbf{0}$. All of this gives us another way of characterising the three types of point described above since at elliptic points $K > 0$ (i.e. κ_1 and κ_2 have the same sign), at hyperbolic points $K < 0$ (i.e. κ_1 and κ_2 have opposite signs) and at parabolic points $K = 0$ (i.e. $\kappa_1 = 0$ and/or $\kappa_2 = 0$). Directions

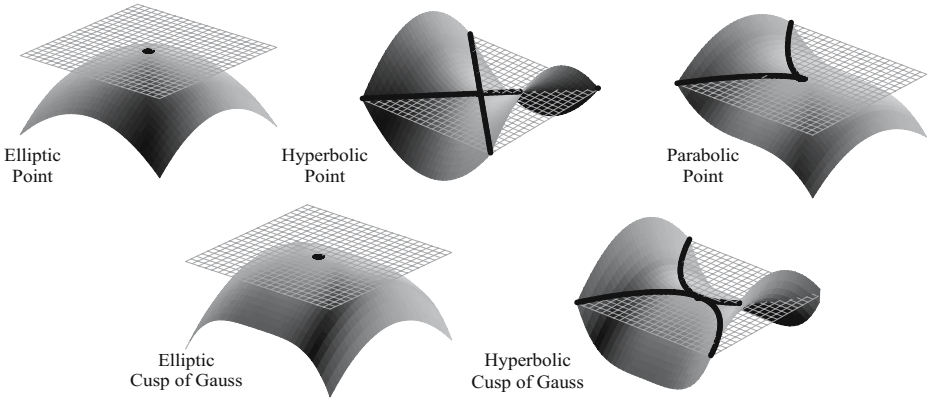


Fig. 1. Contact with tangent plane at elliptic, hyperbolic, parabolic and cusp of Gauss points. The bold black lines show the curve of intersection between the surface and its tangent plane.

in which the sectional curvature is zero are called *asymptotic directions* and if $\kappa_1 = \kappa_2 = 0$ then $\mathbf{0}$ is a degenerate parabolic (or planar) point. Generically the locus of points on M which are parabolic consists of smooth disjoint curves called *parabolic curves*. These curves separate regions of elliptic and hyperbolic points on M and have major significance in computer vision applications, e.g. parabolic points appear as inflexions on the apparent contour of a surface. Parabolic points are clearly points where the tangent plane has especially high *contact* with M. With the surface in Monge form the function f can be regarded as the *height function* on M in the direction $(0, 0, 1)$. We can parametrise all directions close to $(0, 0, 1)$ by $(a, b, 1)$, with a and b both close to zero, so that the family of height functions takes the local form

$$H(x, y; a, b) = (x, y, f(x, y)) \cdot (a, b, 1) = f(x, y) + ax + by.$$

The degree of contact of M with its tangent plane is now measured by singularities of this height function and using Arnold's standard notation for singularities of functions [1] we say f is of type ${A_k}^+$ (resp. ${A_k}^-$) if, by a smooth change of coordinates in x and y, we can reduce to the normal form $x^2 + y^{k-1}$ (resp. $x^2 - y^{k-1}$). More specifically, and with reference to f as given in equation (1):

(i) f is of type A_1 at $\mathbf{0}$ if and only if its quadratic terms do not form a perfect square. This is equivalent to saying that $\mathbf{0}$ is not a parabolic point for which the algebraic condition is ${a_1}^2 \neq 4a_0a_2$. When this holds f is described as a *Morse function* and can be reduced to one of $\pm(x^2 + y^2)$, $x^2 - y^2$. So the zero set of f is locally an isolated point or two transversely intersecting curves, i.e. elliptic and hyperbolic points as described above.

(ii) f is of type $A_{\geq 2}$ at $\mathbf{0}$ if and only if its quadratic terms form a perfect square. Thus $\mathbf{0}$ is a parabolic point, ${a_1}^2 = 4a_0a_2$, and f can be reduced to $a_0x^2 + b_0x^3 + b_1x^2y + b_2xy^2 + b_3y^3 +$ h.o.t. If in addition $a_0 \neq 0$ and $b_3 \neq 0$

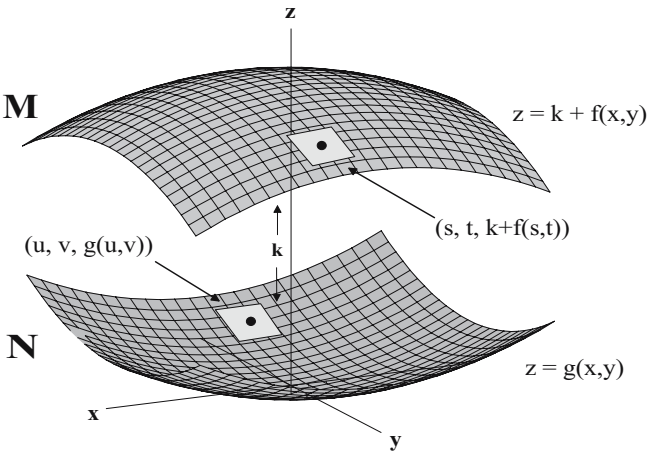


Fig. 2. Geometry of setup for disjoint surface pieces

then f is exactly A_2 and can be reduced to $\pm(x^2 + y^3)$. Thus the zero set of f is locally an ordinary cusp as described above.

(iii) f is of type $A_{\geq 3}$ at $\mathbf{0}$ if and only if its quadratic terms form a perfect square, so that ${a_1}^2 = 4a_0a_2$, but with $a_0 \neq 0$ and $b_3 = 0$. In this case x divides the cubic part of f and it can be reduced to $a_0x^2 + (c_4 - {b_2}^2/4a_0)y^4$. If the coefficient of y^4 here is non-zero then f is exactly A_3. When $b_2 \neq 0$ then $\mathbf{0}$ is called an *ordinary cusp of Gauss* of elliptic or hyperbolic type. For the elliptic type ($A_3{}^+$ with ${b_2}^2 < 4a_0c_4$) the intersection of the surface with its tangent plane is an isolated point whilst for the hyperbolic type ($A_3{}^-$ with ${b_2}^2 > 4a_0c_4$) it is a *tacnode*. The lower half of figure 1 shows these two contact types.

This covers all the generic cases which can occur for a given surface but in a 1-parameter family more complicated situations are possible. At an A_4 we have $c_4 - {b_2}^2/4a_0 = 0$ with the intersection of M and its tangent plane at $\mathbf{0}$ forming a *rhamphoid cusp*. Such points generically represent a more degenerate cusp of Gauss where two ordinary cusps of Gauss momentarily come together. An even more degenerate situation is that of "D_4" points at which the quadratic terms of f vanish completely forming what is termed a *flat umbilic*. We will say something about such non-generic points at the end of the paper but now we go on to describe the geometrical settings for the two cases we wish to study.

1.2 Setup for Disjoint Surface Pieces

Let M and N be disjoint surface pieces in $\mathbb{R}^3$ with parallel tangent planes at p and q respectively. We translate so that q lies at the origin and rotate so that the xy-plane is tangent here. Since parallelism is an affine invariant property we can perform a shear transformation so that p lies on the z-axis (at $z = k$ say). See figure 2. Clearly the tangent plane to M at $(0, 0, k)$ is also parallel to the

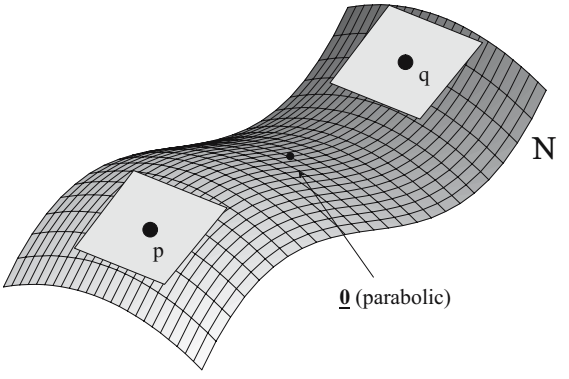


Fig. 3. Geometry of setup for parallel tangents on a single surface piece

xy-plane. Representing M as the graph of a function f of local variables s and t, and N as the graph of a function g of local variables u and v we have, $(s,\, t) \mapsto (s,\, t,\, k + f(s,t))$ and $(u,\, v) \mapsto (u,\, v,\, g(u,v))$ with $f_x = f_y = g_x = g_y = 0$ at $(0,0)$, using x and y to denote differentiation with respect to the first and second local variables respectively. Now, the condition for parallel tangency is simply, $f_x(s,t) = g_x(u,v)$ and $f_y(s,t) = g_y(u,v)$. Thus we can define a map

$$\pi : (s,t,u,v) \mapsto (f_x - g_x\,,\; f_y - g_y) \tag{2}$$

the zeros of which will give us precisely those pairs of points p on M and q on N with parallel tangent planes.

1.3 Setup for the Local Case

Let N be a single surface piece in $\mathbb{R}^3$ with a parabolic point at the origin. We retain the simplifications and notation of the two surface case but now the points $p = (u,\, v,\, g(u,v))$ and $q = (s,\, t,\, g(s,t))$ with parallel tangent planes both lie on N in a neighbourhood of the origin. See figure 3. The map representing N is $(x,\, y) \mapsto (x,\, y,\, g(x,y))$, where g is of the form

$$g(u,v) = u^2 + (g_{30}\,u^3 + g_{21}\,u^2v + g_{12}\,uv^2 + g_{03}\,v^3) + \text{h.o.t.}$$

The parallel tangency condition on N is $g_x(s,t) = g_x(u,v)$ and $g_y(s,t) = g_y(u,v)$, and the zero set of the map $\pi : (s,t,u,v) \mapsto (g_x(s,t) - g_x(u,v),\; g_y(s,t) - g_y(u,v))$ will give us those points p and q with parallel tangent planes on N.

2 Structure of the Set of Parallel Tangent Pairs

We now consider the set $\Pi = \{(s,t,u,v) \in \mathbb{R}^4 : \pi(s,t,u,v) = 0\}$, determine conditions for it to be smooth, investigate ways in which it can be parametrised and study some features of the maps associated with such parametrisations. Due to space constraints detailed proofs of the various propositions stated here are omitted (see [8] for details) but sketch proofs are given in some cases.

2.1 Disjoint Surface Pieces

Recalling the setup described above we take our base-points where $s = t = 0$ and $u = v = 0$ so that all expressions below are evaluated at these points. The Jacobian of π,

$$J_\pi = \begin{pmatrix} f_{xx} & f_{xy} & -g_{xx} & -g_{xy} \\ f_{xy} & f_{yy} & -g_{xy} & -g_{yy} \end{pmatrix}$$

has maximal rank provided some 2×2 minor has non-zero determinant. The first two columns are H_f, the Hessian matrix of f, whilst the last two are (-1 times) H_g, the Hessian matrix of g. Hence Π is certainly smooth when $\det(H_f) \neq 0$ or $\det(H_g) \neq 0$ which is equivalent to either M or N being non-parabolic at $\mathbf{0}$. Moreover we can show that

Proposition 1. *Π is smooth unless both points are parabolic and their unique asymptotic directions are parallel.*

We will consider the case where both points are parabolic later but for now we assume, without loss of generality, that M is non-parabolic. Hence by the implicit function theorem, there exists a map $h : (u, v) \mapsto (s(u, v), t(u, v))$ such that $\pi(s(u, v), t(u, v), u, v) \equiv \mathbf{0}$. The stable possibilities for h (other than it being a local diffeomorphism) are that it be either a *fold* or a *cusp* map, and we now recall some basic properties of these maps.

Fold and Cusp Maps

If the Jacobian matrix of h has non-zero rank at $\mathbf{0}$ then we can change coordinates in the source and target spaces and write $h(u, v) = (u, t(u, v))$, the so-called pre-normal form (note that u, v and t here are not the same as those above, but we will retain these names for simplicity). With the map in this form its Jacobian is given by $J_h = \begin{pmatrix} 1 & 0 \\ t_u & t_v \end{pmatrix}$ whence it is clear that $\det(J_h) = 0 \iff t_v = 0$. Hence the *critical set* of h (i.e. the set of points for which J_h drops below maximal rank) is $\Sigma_h = \{(u, v) : t_v = 0\}$ and by the implicit function theorem this is smooth if and only if $t_{vv} \neq 0$ or $t_{uv} \neq 0$. We can now define fold and cusp maps and state some important equivalence results.

Definition 2. *If $\det(J_h) = 0$ with Σ_h smooth and $h|\Sigma_h$ an immersion at $\mathbf{0}$ then h is called a fold map.*

During the 1950s Whitney proved that any mapping h satisfying these criteria is locally equivalent to the *standard fold map* $h_0 : (x, y) \mapsto (x, y^2)$ in a neighbourhood of $\mathbf{0}$. If we look at the map h_0 we can see where the term 'fold' comes from since if we lift from $\mathbb{R}^2$ to $\mathbb{R}^3$ using $(x, y) \mapsto (x, y, y^2)$ we form a parabolic cylinder containing the x-axis and, as the left half of figure 4 shows, this looks rather like a folded piece of paper. Now $J_{h_0} = \begin{pmatrix} 1 & 0 \\ 0 & 2y \end{pmatrix}$ so that $\det(J_{h_0}) = 0$ if and only if $y = 0$ and $\Sigma_{h_0} = \{(x, y) : y = 0\}$. Hence the critical set of the map

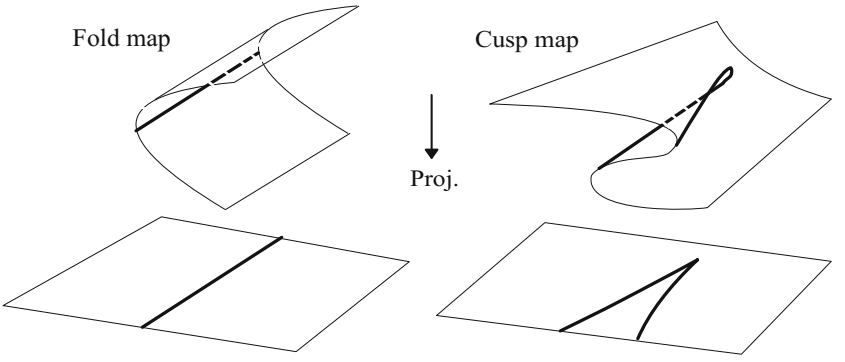


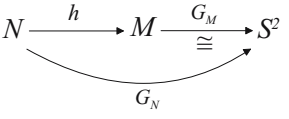
Fig. 4. Fold and Cusp maps. The dark line in the upper figures represents the *contour generator* (or *critical set*) of the vertical projection and, in the lower figures, the *apparent contour* or *critical locus* of the projection.

h_0 is just the x-axis. This is as we expect since Σ_{h_0} can be thought of as the contour generator of the lifted surface viewed (in this instance) from $y = +\infty$. Also $h_0|\Sigma_{h_0} = (x, 0)$ so the apparent contour is smooth.

Definition 3. *If* $\det(J_h) = 0$ *with* Σ_h *smooth but* $h|\Sigma_h$ *an ordinary cusp at* $\mathbf{0}$ *then* h *is called a cusp map.*

In this case Whitney proved that any such h is equivalent to the map $h_1 : (x, y) \mapsto (x, xy - y^3)$ in a neighbourhood of $\mathbf{0}$. Lifting to $\mathbb{R}^3$ using $(x, y) \mapsto (x, y, xy - y^3)$ we form a surface rather like a twisted plane, as shown in the right half of figure 4. Now $J_{h_1} = \begin{pmatrix} 1 & 0 \\ y & x - 3y^2 \end{pmatrix}$ so that $\det(J_{h_1}) = 0$ if and only if $x = 3y^2$ and $\Sigma_{h_1} = \{(x, y) : x = 3y^2\}$. Hence the critical set of the map h_1 is represented by a parabola through the origin and viewing the lifted surface from $y = +\infty$ gives an ordinary cusp at the origin as an apparent contour, hence the name. Also $h_0|\Sigma_{h_0} = (3y^2, 4y^3)$, which is clearly an ordinary cusp.

Now, consider the *Gauss map* of N, i.e. the map $G_N : N \mapsto S^2$ where $p \mapsto n_p$ the oriented unit normal at p. For this map it is standard theory that (i) G_N is a local diffeomorphism if and only if p is non-parabolic, (ii) G_N is a fold map if and only if p is parabolic but not a cusp of Gauss, and (iii) G_N is a cusp map if and only if p is an ordinary cusp of Gauss. So if G_M is the Gauss map of M then it is a local diffeomorphism (since we assume M to be non-parabolic). The map h relates points on N to points on M with parallel tangent planes and so we have the following equivalence of maps,



since the unit normal on N maps to the same unit normal on M under h. Hence $h \equiv {G_M}^{-1} \circ G_N$ and since ${G_M}^{-1}$ is also a local diffeomorphism then h and G_N have identical singularities giving the following

Proposition 4. *The map $h : (u, v) \mapsto (s, t)$ is: (i) a fold map if and only if M is non-parabolic at $(0, 0, k)$ and N is parabolic but not a cusp of Gauss at $(0, 0, 0)$, (ii) a cusp map if and only if M is non-parabolic at $(0, 0, k)$ and N is an ordinary cusp of Gauss at $(0, 0, 0)$.*

When both points are parabolic we cannot parametrise Π using only the parameters of one of the points (i.e. (s, t) or (u, v)). However, we can express t and u as smooth functions of s and v and consider the pair of maps $F_1 : (s, v) \mapsto (s, t(s, v))$ and $F_2 : (s, v) \mapsto (u(s, v), v)$. It turns out that if neither parabolic point is a cusp of Gauss then F_1 and F_2 are fold maps and that there is a smooth curve lying in each surface piece which separates points which are candidates for parallel tangency from those which are not. By proposition 1 we know that Π is smooth here provided the asymptotic directions at the two parabolic points are not parallel.

2.2 The Local Case

With just a single surface piece we are faced with a problem that cannot arise in the disjoint surfaces case, namely that points in the diagonal $\Delta = \{(s, t, u, v) : s = u \text{ and } t = v\}$ are always a solution of the parallel tangency equations (since they give identical points of N). In order to proceed we need to exclude such solutions. Taking g in Monge form then

$$g(x, y) = x^2 + (b_0\, x^3 + b_1\, x^2 y + b_2\, x y^2 + b_3\, y^3) + \text{h.o.t.} \tag{3}$$

so that if $b_3 = 0$ then $\mathbf{0}$ is a cusp of Gauss on N. If we let $F = g_x(s, t) - g_x(u, v)$ and $G = g_y(s, t) - g_y(u, v)$ then $F = (\, 2s + 3b_0 s^2 + 2b_1 st + b_2 t^2 + \text{h.o.t.}\,) - (\, 2u - 3b_0 u^2 - 2b_1 uv - b_2 v^2 + \text{h.o.t.}\,) \equiv 0$ and since s has a non-zero coefficient it follows from the implicit function theorem that we can write $s = s(t, u, v)$ (similarly we could write $u = u(s, t, v)$). If $G_1(t, u, v) = G(s(t, u, v), t, u, v)$ then we find that $G_1(t, u, v) = (t - v)\,(3b_3 t + 2b_2 u + 3b_3 v + \text{h.o.t.}) \equiv 0$ whence $H(t, u, v) = 3b_3 t + 2b_2 u + 3b_3 v + \text{h.o.t.} \equiv 0$ represents the parallel tangent set with the diagonal points removed. If $b_3 \neq 0$ then t or v can be expressed uniquely as a smooth function of the other two variables. If we write $t = t(u, v)$ then calculations give $h(u, v) = (u + \text{h.o.t.},\, -2b_2\, u/(3b_3) - v + \text{h.o.t.})$ hence we have

Proposition 5. *The map $h : (u, v) \mapsto (s, t)$, which associates to each point of N near $\mathbf{0}$ the unique (distinct) point at which the tangent planes are parallel, is a local diffeomorphism when $\mathbf{0}$ is parabolic but not a cusp of Gauss.*

Ordinary Cusp of Gauss

We recall from our earlier definitions that $b_3 = 0$, $b_2 \neq 0$ and ${b_2}^2 \neq 4c_4$ at an ordinary cusp of Gauss, the coefficients here referring to those of equation (3). It

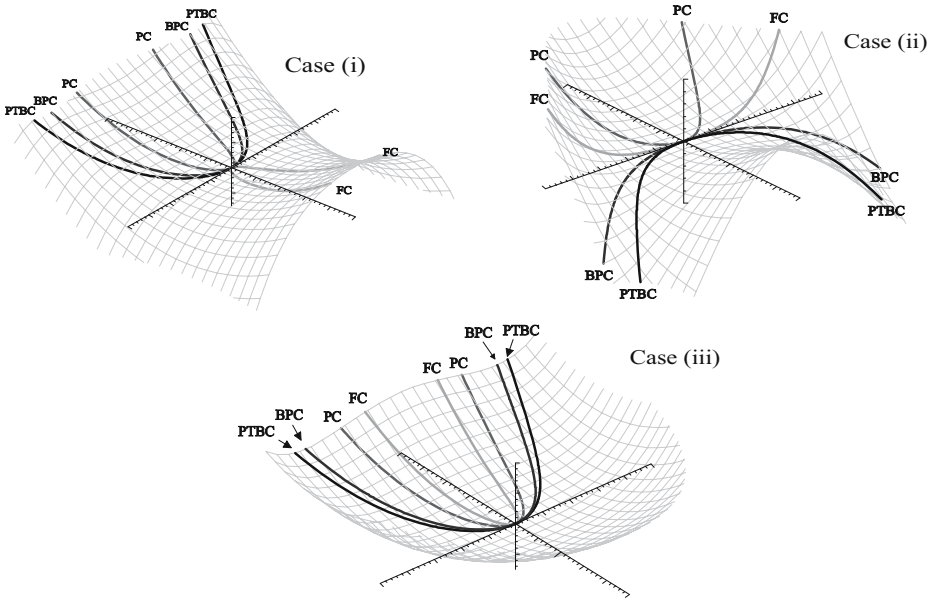


Fig. 5. Arrangement of Special Curves through an Ordinary Cusp of Gauss

turns out that Π is smooth in this case but, as with both points parabolic for disjoint surfaces, we cannot find one point of parallel tangency as a function of the other. Instead we write s and u as smooth functions of t and v and consider the maps $F_1 : (t, v) \mapsto (s(t, v), t)$ and $F_2 : (t, v) \mapsto (u(t, v), v)$. In a similar manner to the both points parabolic for disjoint surfaces case we can show that these are both fold maps. The common image of F_1 and F_2 restricted to their respective critical sets represents the boundary of the region on N which contains candidate points for parallel tangency. We call this curve the Parallel Tangents Boundary Curve (PTBC) and it is interesting to describe the relationship between it and some other important affinely invariant curves which pass through the cusp of Gauss on N. The curves that we consider are: the Parabolic Curve (PC), the *Flecnodal Curve* (FC) which is the locus of points on M where some tangent line has 4-point (or higher) contact and, the *Bitangent Plane Curve* (BPC) which is the locus of points p on N for which there exists a distinct point q with *the same* tangent plane. Some simple inequality calculations demonstrate that these curves lie relative to one another in three distinct ways depending on the value of the quotient ${c_4}/{{b_2}^2}$ of coefficients from equation (3) as follows:

(i) ${c_4}/{{b_2}^2} < -\frac{1}{12}$. Here the PC and PTBC are always on the same side of the y-axis. Within this region the BPC lies on the same side of the y-axis as the PC but the FC always lies on the opposite side of the y-axis from the PC, e.g. $g(x, y) = x^2 + xy^2 - y^4$ where the PTBC and BPC lie in the hyperbolic region. See figure 5 (top left).

(ii) $-\frac{1}{12} < {c_4}/{{b_2}^2} < \frac{1}{6}$. Here the PC and PTBC are always on opposite sides of the y-axis. Within this region the FC and BPC could lie on either side of the y-axis, e.g. $g(x,y) = x^2 + 4xy^2 + y^4$ where the PTBC and BPC lie in the hyperbolic region. See figure 5 (top right).

(iii) $\frac{1}{6} < {c_4}/{{b_2}^2}$. Here the PC and PTBC are always on the same side of the y-axis. Within this region the FC and BPC both lie on the same side of the y-axis as the PC, e.g. $g(x,y) = x^2 + xy^2 + y^4$ where the PTBC and BPC lie in the elliptic region. See figure 5 (bottom).

* The reader is referred to Uribe-Vargas [7] for further properties and a more extensive treatment of cusps of Gauss, including further examples of the significance of the important quotient ${c_4}/{{b_2}^2}$.

3 The Mid-Point Tangent Surface

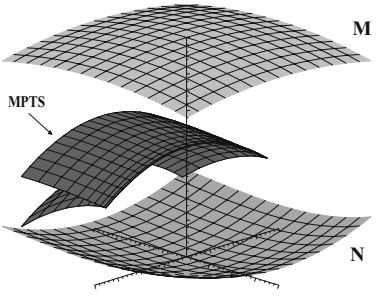
We now consider the *equidistants* formed by points at a fixed position along chords joining parallel tangent pairs. These are affine invariants of the pair of surfaces or single surface piece. If the points of parallel tangency are at p and q then the equidistant point is given by $(1-\lambda)\,p + \lambda\,q$ where $\lambda \in (0,1)$ is the fixed proportion along the chord. Note that λ and $1-\lambda$ will give the same equidistant (since p and q can be interchanged) so when $\lambda = 1/2$ the equidistant has a special symmetry. We call this particular equidistant the Mid-Point Tangent Surface (or MPTS) and concentrate on its properties throughout this section.

3.1 Disjoint Surface Pieces

Using X to denote the MPTS and its local parametrisation we will first determine conditions it to be a smooth mapping. Assuming that Π is smooth (by proposition 1) then a standard result says that X is smooth provided the 4×5 matrix with first three rows J_X and last two rows J_π has maximal rank. This reduces to asking when $(f_{ss} + g_{uu})(f_{tt} + g_{vv}) - (f_{st} + g_{uv})^2 \neq 0$. If H_f is the Hessian of f and H_g is the Hessian of g then this condition is equivalent to $\det(H_f + H_g) \neq 0$ hence we have

Proposition 6. *The MPTS is smooth at a point p when the sum of the hessian matrices of M and N, evaluated at the end points of the chord which formed p, is non-singular.*

As an example we take $f(s,t) = -s^2 - t^2$, $k = 3$ and $g(u,v) = u^2 + 2v^2 + u^3$, so M and N are elliptic at $(0,0,3)$ and $(0,0,0)$ respectively. The MPTS fails to be smooth at $(0,0,3/2)$ since $(H_f + H_g)(\mathbf{0}) = \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix}$ which is clearly singular. The figure below shows the two surface pieces M and N (in light) with the MPTS (in dark) lying in-between.



Discriminants and Unfoldings

We can show that the tangent plane at any smooth point of an equidistant is parallel to the tangent planes on M and N which gave rise to it. Thus equidistants can also be formed as the envelope of a certain family of planes, making it possible to perform unfolding calculations and so classify some of their singularities. We sketch the ideas here for the MPTS but more detail and consideration of other equidistants can be found in [8, Chap 3]. For the MPTS we take the following family as a *generating function*

$$F(s,t,u,v,p,q;x,y,z) = f(s,t) + g(u,v) + k - 2z + (s+u-2x)\,p + (t+v-2y)\,q$$

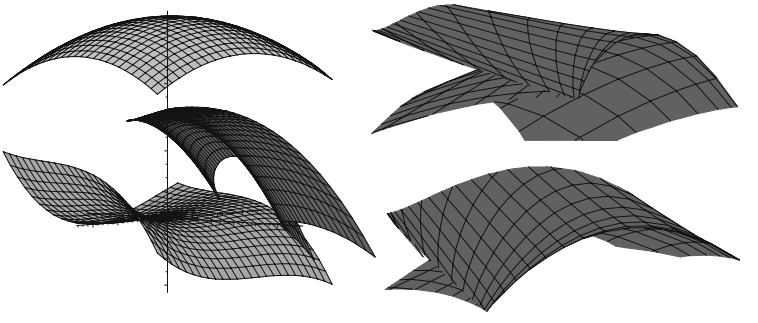
where $(p,q,1)$ is an arbitrary direction vector at the origin. Some simple calculations show that the *discriminant* of this family, $\mathcal{D}_F = \{(x,y,z) : F = F_s = F_t = F_u = F_v = F_p = F_q = 0 \text{ for some } s,t,u,v,p,q\}$, is precisely the MPTS. We can show that $F^{-1}(\mathbf{0})$ is smooth at $(0,\ldots,0,k/2)$ in $\mathbb{R}^9$ provided either (i) M or N is non-parabolic, or (ii) both M and N are parabolic but with distinct asymptotic directions (i.e. the same conditions as for Π to be smooth by proposition 1). Thus a singularity of the MPTS at $(0,0,k/2)$ will arise only as a result of the projection of $F^{-1}(\mathbf{0})$ to $\mathbb{R}^3$. We can reduce the number of variables in F from six to two, greatly simplifying the unfolding calculations (see Giblin and Zakalukin [3] for details) and by a series of smooth changes of variable and elimination of a non-degenerate quadratic form we can reduce to

$$H(r_1, r_2\,;\, x, y, z) = f(r_1 + x,\, r_2 + y) + g(-r_1 + x,\, -r_2 + y) - 2z + k$$

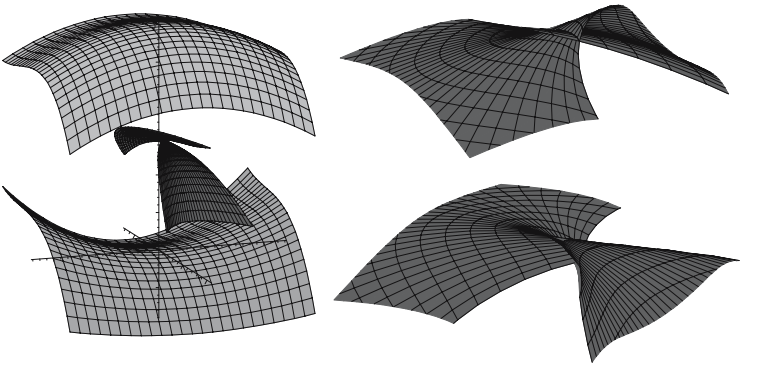
where F and H have diffeomorphic discriminants. We now use H, a three parameter family of functions of two variables, in the unfolding calculations. Taking a Monge form for N, so that $g(x,y) = (g_{20}\,x^2 + g_{11}\,xy + g_{02}\,y^2) + (g_{30}\,x^3 + g_{21}\,x^2y + g_{12}\,xy^2 + g_{03}\,y^3) + \text{h.o.t.}$ and rotating in the $z-$axis so that the principal directions on M align with the x and y axes then, $f(x,y) = (a\,x^2 + b\,y^2) + (f_{30}\,x^3 + f_{21}\,x^2y + f_{12}\,xy^2 + f_{03}\,y^3) + \text{h.o.t.}$ Now, standard methods of singularity theory (e.g. Poston and Stewart [6], Chap 8) enable us to find the local structure of the MPTS by determining singularity types and working out versal unfolding criteria. The exact conditions for A_2 and A_3 singularities of the

MPTS have been determined and details can be found in [8, Chap 3], but here we restrict ourselves to some examples which satisfy these conditions:

(i) For the A_2 case we take $k = 3$, $f(x, y) = -x^2 - y^2$ and $g(x, y) = -x^2 + y^2 + x^3 + xy^2 + y^3$, obtaining the following figure which shows three views of a cusp edge on the MPTS passing through $(0, 0, 3/2)$.



(ii) For the A_3 case we take $k = 3$, $f(x, y) = -x^2 - y^2 + x^3$ and $g(x, y) = -x^2 + y^2 + x^3 - xy^2$, obtaining the following figure which shows three views of a swallowtail point on the MPTS at $(0, 0, 3/2)$.



The versal unfolding criteria for A_2 and A_3 singularities both require at least one of the surface pieces to be non-parabolic. However when M and N are both parabolic (but neither is a cusp of Gauss) we can write t and u as smooth functions of s and v and the map $h : (s, v) \mapsto (t, u)$ we know to be a fold (from above). Calculating the components of X in this case we obtain, $X(s, v) = (s/2 + \text{h.o.t.}\,, (v - as)/2 + \text{h.o.t.}\,, k/2 + \text{h.o.t.})$ from which it is clear that J_X has maximal rank when $s = v = 0$. Hence X is always smooth in this case.

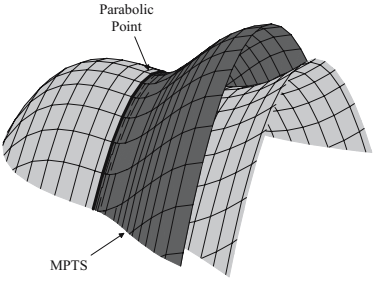
3.2 The Local Case

Here we consider pairs of points with parallel tangent planes close to a parabolic point on a single surface piece. In section 2.2 we showed that the parallel tangency

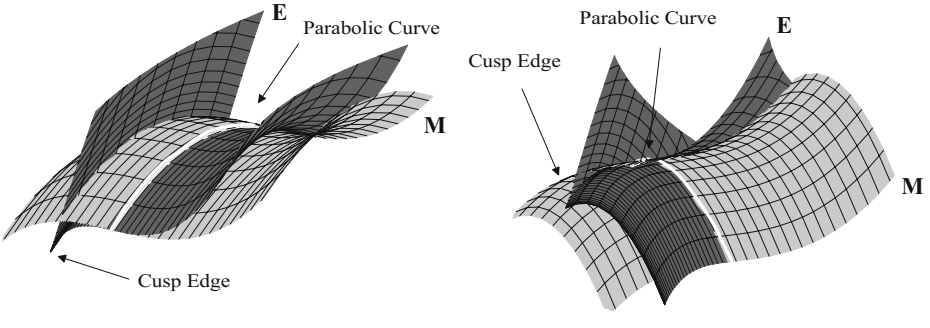
map π of equation (2) was a local diffeomorphism provided the parabolic point at $\mathbf{0}$ is not a cusp of Gauss. So we can find s and t as functions of u and v whence the MPTS is described in the normal way as

$$X(u,v) = \frac{1}{2}\,(s(u,v)+u,\, t(u,v)+v,\, f(s(u,v),t(u,v))+f(u,v))\,.$$

If we do a calculation using f in the form of equation (3) and look for singular points by postulating a curve lying in the MPTS (close to $\mathbf{0}$) at all points of which J_X drops below maximal rank we find that the only such curve is the parabolic curve of the original surface. So in the local case the MPTS consists of a smooth sheet terminating along the parabolic curve, e.g. with $f = x^2+y^3+y^4$ the MPTS appears as follows.



We can gain further insight here by considering the other equidistants which all cut the surface inflectionally along the parabolic curve then turn back along a cusp edge, e.g. taking $f = x^2+y^3+y^4$ as above but with $\lambda = 1/4$ the equidistant E appears as follows.



So it is clear that the MPTS is the limiting case of an equidistant which stops on the parabolic curve before retracing its path. It is natural to ask if the MPTS can be regarded as a discriminant in the local case, and if so can we find an appropriate generating function for it, analogous to the functions F and H above? In trying to answer this we are again faced with the problem that the two points with parallel tangent planes can come into coincidence. Including such solutions of the parallel tangency equations means that any family of functions

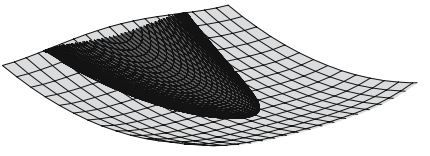
we choose will include the original surface as part of its discriminant and so we need to find a way of excluding them. Again we omit the details here but we find that this can be achieved by using the family

$$G(s_1, s_2, h; z_1, z_2) = f(s_1 + z_1, s_2 + z_2) + f(s_1 - z_1, s_2 - z_2) - 2h$$

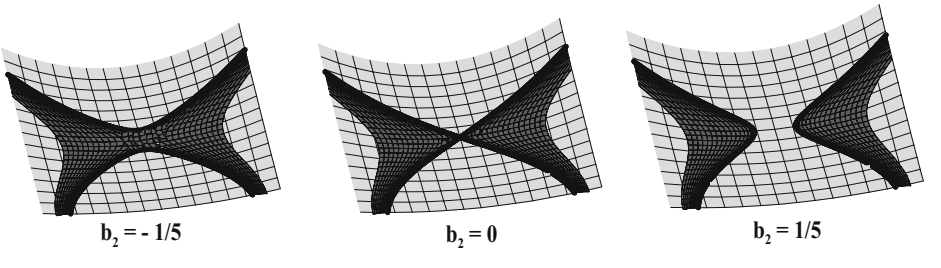
and factoring out the solution $z_2 = 0$ (which represents the original surface since we find that $z_2 = 0 \implies z_1 = 0$). We are left with a function $K(s_1, s_2, z_2) = 0$, not divisible by z_2, which together with $G = 0$ allows us to parametrise the MPTS using s_1 and z_2. All other equidistants can be parametrised more simply using s_1 and s_2.

Cusps of Gauss

With the surface described in the form of equation (3) then the origin is a cusp of Gauss when $b_3 = 0$. In this case we find that $K(s_1, s_2, z_2) = 4\,b_2\,s_1 - 8\,b_1\,b_2\,s_1\,s_2 + 4\left(6\,c_4 - {b_2}^2\right){s_2}^2 + 4\left(c_2 - {b_1}^2\right){s_1}^2 + 8\,c_4\,{z_2}^2 + \text{h.o.t.}$ so we can write s_1 uniquely as a function of s_2 and z_2 if and only if $b_2 \neq 0$. Thus if $b_3 = 0$ but $b_2 \neq 0$ then the MPTS remains a smooth surface with boundary along the parabolic curve, e.g. with $f = x^2 + xy^2 + y^4$ the MPTS appears as follows.



If $b_2 = 0$ then $K(s_1, s_2, z_2) = 24\,c_4\,{s_2}^2 + 4\left(c_2 - {b_1}^2\right){s_1}^2 + 8\,c_4\,{z_2}^2 + \text{h.o.t.}$ If c_4 and $c_2 - {b_1}^2$ have the same sign then this is locally an isolated point in (s_1, s_2, z_2)–space, whilst if they have opposite sign it is locally a cone. When $b_2 = 0$ the parabolic curve is singular and as the zero set of a Morse function this singular point will be locally a point or a transverse crossing at the origin. In a generic family, as b_2 passes through zero, we observe the standard Morse transition with the MPTS clinging to the parabolic curve throughout, e.g. with $f = x^2 + b_2\,x^2\,y + y^4$ we observe the following.



* The reader is referred to Holtom [4] for a treatment of equidistants in the plane, and M. van Manen [5] for the more general n-dimensional case.

4 Conclusions and Further Work

We have obtained a number of results concerning maps linking surface points with parallel tangent planes for disjoint surface pieces and for the local case, in the neighbourhood of a parabolic point. We have used these to study affinely invariant symmetry constructions called equidistants, formed by points at a fixed proportion along chords joining parallel tangent pairs. We paid particular attention to the half way equidistant (or MPTS) and for disjoint surface pieces obtained exact conditions for cusp edge and swallowtail singularities. In the local case we demonstrated various differences in behaviour between the MPTS and all other equidistants.

The work on equidistants in the local case is preliminary and we are interested in the types of singularity that equidistants other than the MPTS can display. For the MPTS we would like to show how its structure varies as the parabolic curve passes through various transformations in a 1-parameter family as described in Bruce et al [2]. We showed above how the MPTS behaves during a Morse transition of the parabolic curve and we are currently looking at its behaviour during more degenerate transitions, e.g. transitions about A_4 and D_4 points.

Acknowledgments

The author was funded by the EPSRC during the preparation of this paper.

References

1. Arnol'd, V.I.: Catastrophe Theory (Translated from Russian). Springer, Heidelberg (1984)
2. Bruce, J.W., Giblin, P.J., Tari, F.: Parabolic curves of evolving surfaces. International Journal of Computer Vision 17, 291–306 (1996)
3. Giblin, P.J., Zakalukin, V.: Singularities of Centre Symmetry Sets. Proceedings of the London Mathematical Society 90, 132–166 (2005)
4. Holtom, P.A.: Affine Invariant Symmetry Sets. PhD Dissertation, Liverpool (2001), Available on `http://www.liv.ac.uk/~pjgiblin`
5. van Manen, M.: Conflict sets, Orthotomics, Pedals and Billiards as Canonical Relations. Math. Proc. Cambridge Philos. Soc. 141, 339–353 (2006)
6. T. Poston, T., Stewart, I.: Catastrophe Theory and its Applications. Pitman (1978)
7. Uribe-Vargas, R.: A projective invariant for swallowtails and godrons and theorems on the flecnodal curve. Moscow Math. J. 6, 731–768 (2006)
8. Warder, J.P.: PhD Dissertation, University of Liverpool, (in preparation)

Condition Numbers and Least Squares Regression

Joab R. Winkler

Department of Computer Science, The University of Sheffield
Regent Court, 211 Portobello Street, Sheffield S1 4DP, UK
j.winkler@dcs.shef.ac.uk

Abstract. Many problems in geometric modelling require the approximation of a set of data points by a weighted linear combination of basis functions. This yields an over-determined linear algebraic equation, which is usually solved in the least squares (LS) sense. The numerical solution of this problem requires an estimate of its condition number, of which there are several. These condition numbers are considered theoretically and computationally in this paper, and it is shown that they include a simple normwise measure that may overestimate by several orders of magnitude the true numerical condition of the LS problem, to refined componentwise and normwise measures. Inequalities that relate these condition numbers are established, and it is concluded that the solution of the LS problem may be well-conditioned in the normwise sense, even if one of its components is ill-conditioned. An example of regression using radial basis functions is used to illustrate the differences in the condition numbers.

1 Introduction

Many problems in geometric modelling require the construction of a smooth curve that approximates a set of data points. The number of basis functions that define the approximating curve is usually smaller than the number of data points, and thus this approximation problem yields the least squares (LS) problem

$$\min_x \|Ax - b\|, \qquad \|\cdot\| = \|\cdot\|_2, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m \geq n$ and rank $A = n$. It is assumed that the basis functions, and therefore the elements of A, are known exactly, and that there exist measurement errors in the right hand side vector b. In this paper, four condition numbers of the LS problem (1) are considered theoretically and computationally. The simplest and least refined condition number is the normwise condition number

$$\kappa_{LS}(A, \theta) = \kappa(A)/\cos\theta,$$

where $\kappa(A)$ is equal to the ratio of the largest singular value to the smallest singular value of A, and $\cos\theta \neq 1$ if b does not lie in the column space of A.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 480–493, 2007.
© Springer-Verlag Berlin Heidelberg 2007

By contrast, the most refined condition numbers are obtained by assigning a condition number that is a function of b to each component of the vector x_0 that minimises (1).

The vector x_0 is the solution of

$$A^T Ax = A^T b,$$

and is given by

$$x_0 = A^\dagger b = VS^\dagger U^T b = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i,$$

where $X^\dagger = \left(X^T X\right)^{-1} X^T$ is the left inverse of X, USV^T is the singular value decomposition (SVD) of A, v_i is the i'th column of V, u_i^T is the i'th row of U, and $\sigma_i, i = 1, \ldots, n$, are the singular values of A, arranged in non-increasing order.

Four condition numbers of the LS problem are considered in Section 2, and it is shown in Section 3 that they are related by a series of inequalities. Several examples that illustrate the results are given, and Section 4 contains a summary of the paper.

2 Condition Numbers for the Least Squares Problem

Four condition numbers of the LS problem are considered in this section. They range from a simple normwise measure to refined componentwise measures, and an example of regression using radial basis functions is used to illustrate the disadvantage of the simple normwise measure. The existence of several condition numbers of an equation is not limited to the LS problem. For example, componentwise and normwise, and average case and worst case, condition numbers of a root of a polynomial are considered in [5], and the computational results show that significant differences exist between these measures.

Componentwise condition numbers of the LS problem may be important because the components of the vector x_0 of a problem in regression are the coefficients of the basis functions. If it is required, for example, to compute the roots of the regression curve, knowledge of the error bounds of each of these coefficients is required to determine the error bounds of each root.

2.1 A Simple Normwise Condition Number

It is shown in [4] that a simple normwise condition number of x_0 is

$$\kappa_{LS}(A, \theta) = \max_{\delta b, b \in \mathbb{R}^m} \frac{\Delta x_0}{\Delta b} = \frac{\sigma_1}{\sigma_n} \frac{1}{\cos\theta} = \frac{\kappa(A)}{\cos\theta}, \tag{2}$$

where

$$\Delta x_0 = \frac{\|\delta x_0\|}{\|x_0\|}, \qquad \Delta b = \frac{\|\delta b\|}{\|b\|},$$

the residual r and angle θ satisfy

$$r := Ax_0 - b, \qquad \|r\| = \|b\| \sin\theta \qquad \text{and} \qquad \|Ax_0\| = \|b\| \cos\theta, \tag{3}$$

and $\kappa(A) = \sigma_1/\sigma_n$ is the condition number of A.

Practical problems rarely generate situations in which $\theta \approx \pi/2$ because it is usual to pose the LS problem such that b can be approximated, with a small error, by a linear combination of the columns of A. The matrix A incorporates, for example, the basis functions that are used in regression, and a poor choice of basis functions may lead to a large value of θ. Even if θ is small, $\kappa_{LS}(A, \theta)$ is large if A is ill-conditioned.

The main disadvantage of $\kappa_{LS}(A, \theta)$ follows immediately from (2). In particular, it is a worst case upper bound of the true numerical condition of x_0, taken over all vectors b and δb, and the following example shows that $\kappa_{LS}(A, \theta)$ may overestimate the true numerical condition by several orders of magnitude.

Example 1. Regression was performed on two sets of 100 data points $(x_i, y_i)\,, i = 1, \ldots, 100$, in the interval $I = [1, \ldots, 20]$ using a linear combination of 33 radial basis functions,

$$y_i = \sum_{k=1}^{33} a_k \exp\left(-\frac{(x_i - d_k)^2}{2\sigma_d^2}\right), \qquad i = 1, \ldots, 100,$$

where $\sigma_d = 1.35$ and the centres d_k of the basis functions are not uniformly distributed in I. The coefficient matrix A is therefore of order 100×33, b stores the function values y_i, and x stores the coefficients a_k of the radial basis functions.

The first set of 100 points, which are stored in the vector $b = b_1$, and their regression curve are shown in Figure 1, and it is seen that the error between the exact data points and the approximating curve is small. The LS problem was solved twice, once for the exact data $b = b_1$, and once for perturbed data $b_1 + \delta b_1$, where the elements of δb_1 are drawn from a zero mean Gaussian distribution, such that $\|b_1\| \,/\, \|\delta b_1\| = 2.9 \times 10^5$. The solutions of the LS problem for these problems are shown in Figure 2, and it is seen that the noise has a significant effect on the coefficients of the radial basis functions, such that the solution in the presence of noise has no value.

The experiment was repeated for the second set of 100 data points, which are stored in the vector $b = b_2$. These points are shown in Figure 3, which also shows the regression curve that is obtained using the same matrix A because the same radial basis functions were used. The LS problem was therefore solved twice, once for the exact data $b = b_2$, and once for perturbed data $b_2 + \delta b_2$ where $\|b_2\| \,/\, \|\delta b_2\| = 1.2 \times 10^3$. Figure 4 shows the coefficients of the regression curve for these two cases, and it is seen that the noise has very little effect on the exact coefficients.

The condition number of A is $\kappa(A) = 5.1 \times 10^8$ and the vectors b_1 and b_2 lie in the column space of A, from which it follows that $\kappa_{LS}\,(A, \theta) = \kappa(A)$ for $b = b_1$ and $b = b_2$. Figures 2 and 4 show that this measure fails to distinguish between the vectors b_1 and b_2, that is, the vectors b for which the LS problem is ill-conditioned, and the vectors b for which this problem is well-conditioned. □

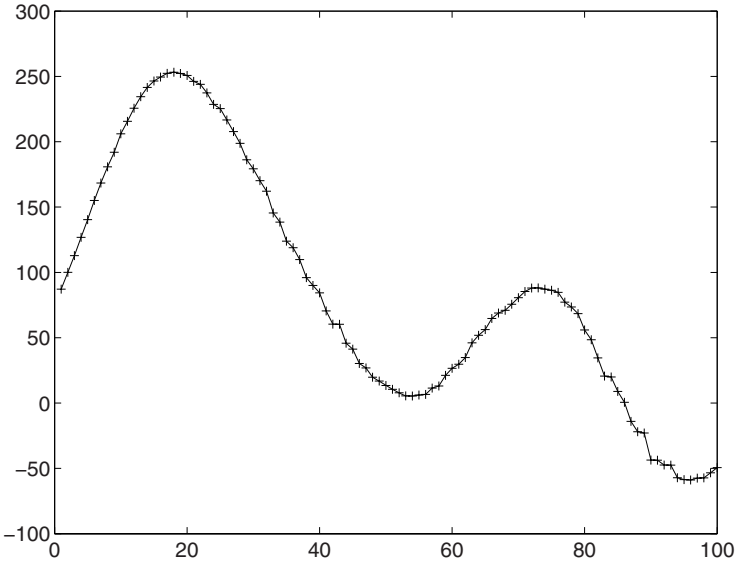


Fig. 1. A set of 100 points and their approximating curve for $b = b_1$

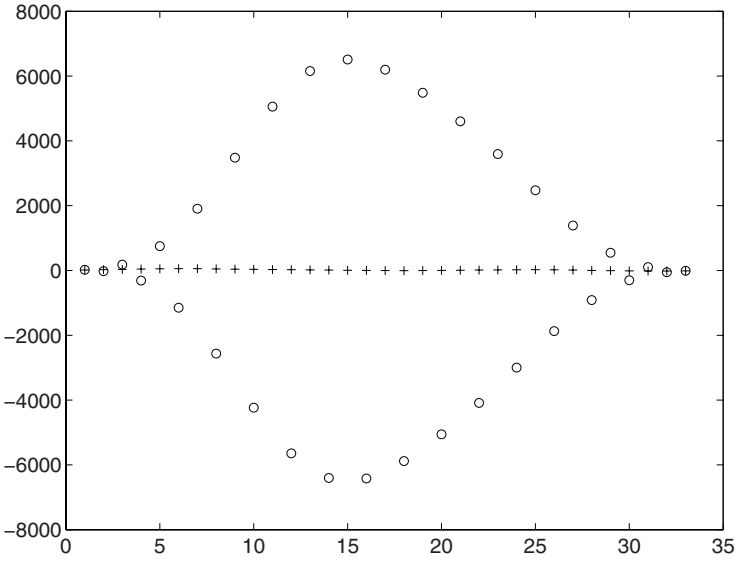


Fig. 2. The coefficients of the radial basis functions for the exact (+) and noisy (o) data for the curve shown in Figure 1

The failure of $\kappa_{LS}(A, \theta)$ to quantify the true numerical condition of the LS problem is a significant disadvantage that may lead to incorrect conclusions. For

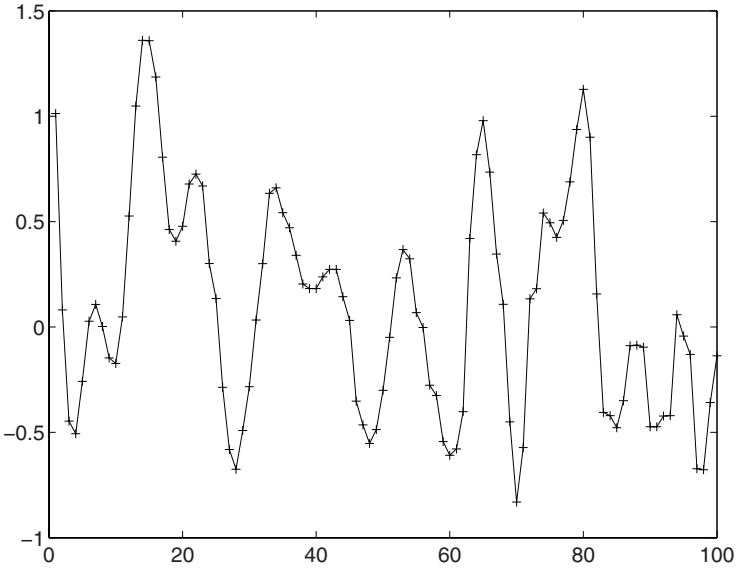


Fig. 3. A set of 100 points and their approximating curve for $b = b_2$

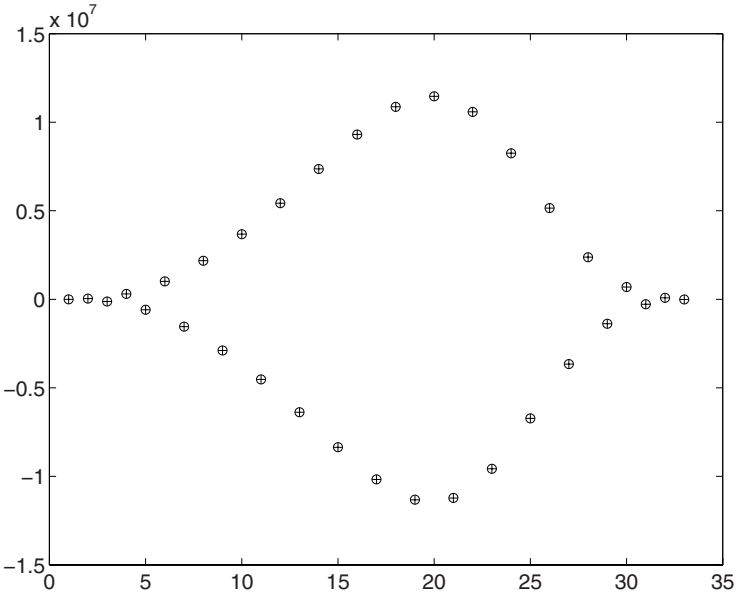


Fig. 4. The coefficients of the radial basis functions for the exact (+) and noisy (o) data for the curve shown in Figure 3

example, since $\cos\theta = 1$, it follows that $\kappa_{LS}\,(A, \theta) = \kappa(A) = 5.1 \times 10^8$, and thus it may be thought that regularisation must be applied to the LS problem for both

$b = b_1$ and $b = b_2$ in order to achieve computationally reliable solutions. This procedure will, however, lead to a totally incorrect result for the LS problem for $b = b_2$, and its correct application when $b = b_1$ requires that prior information be known [2]. This disadvantage of $\kappa_{LS}(A, \theta)$ is overcome by the effective condition number.

2.2 The Effective Condition Number

The effective condition number $S_{LS}(A, b)$ is superior to the condition number $\kappa_{LS}(A, \theta)$ because it distinguishes between the vectors b for which the LS problem is ill-conditioned, and the vectors b for which the LS problem is well-conditioned. It is defined as the maximum value of the ratio $\Delta x_0/\Delta b$ over all perturbations $\delta b \in \mathbb{R}^m$,

$$S_{LS}(A, b) := \max_{\delta b \in \mathbb{R}^m} \frac{\Delta x_0}{\Delta b} = \frac{1}{\sigma_n} \frac{\|b\|}{\|x_0\|} = \frac{1}{\sigma_n} \frac{\|b\|}{\|S^\dagger U^T b\|}, \tag{4}$$

and it satisfies the inequality

$$1 \leq S_{LS}(A, b) \leq \kappa_{LS}(A, \theta). \tag{5}$$

The lower bound in (5) follows immediately, and derivation of the upper bound is simplified by the introduction of the vector $c \in \mathbb{R}^m$,

$$c = \{c_i\}_{i=1}^m = U^T b,$$

from which it follows that (4) can be written as

$$S_{LS}(A, Uc) = \frac{\|c\|}{\sigma_n \sqrt{\sum_{i=1}^n \left(\frac{c_i}{\sigma_i}\right)^2}}.$$

Since

$$\sigma_n \sqrt{\sum_{i=1}^n \left(\frac{c_i}{\sigma_i}\right)^2} = \frac{\sigma_n}{\sigma_1} \sqrt{\sum_{i=1}^n \left(\frac{\sigma_1}{\sigma_i} c_i\right)^2} \geq \frac{1}{\kappa(A)} \sqrt{\sum_{i=1}^n c_i^2},$$

it follows that

$$S_{LS}(A, Uc) \leq \kappa(A) \frac{\|c\|}{\sqrt{\sum_{i=1}^n c_i^2}}. \tag{6}$$

The application of the SVD of A to (3) shows that

$$\cos\theta = \frac{\|Ax_0\|}{\|b\|} = \frac{\left\|SS^\dagger c\right\|}{\|c\|} = \frac{\sqrt{\sum_{i=1}^n c_i^2}}{\|c\|},$$

and thus the upper bound in (5) follows from (6).

Example 2. Consider Example 1 in which the LS problem was considered for two right hand side vectors b. The effective condition numbers of this problem for these vectors are

$$S_{LS}(A, b_1) = 4.6 \times 10^8 \qquad \text{and} \qquad S_{LS}(A, b_2) = 7.9.$$

It is clear that these values are consistent with Figures 2 and 4, respectively, and thus the effective condition number is a measure, in the normwise sense, of the numerical condition of the LS problem. □

It has been shown that the effective condition $S_{LS}(A, b)$ is superior to the condition number $\kappa_{LS}(A, \theta)$ because more accurate information on the true numerical condition of the LS problem can be determined. It would appear, therefore, that the disadvantage of $\kappa(A)$ and $\kappa_{LS}(A, \theta)$ has been overcome. The following example shows, however, that more refined condition numbers must be developed.

Example 3. Consider the matrix A and vector b that are given by [1],

$$A = \begin{bmatrix} 0.4919 & 0.1112 & -0.6234 & -0.6228 \\ -0.5050 & -0.6239 & 0.0589 & 0.0595 \\ 0.5728 & -0.0843 & 0.7480 & 0.7483 \\ -0.4181 & 0.7689 & 0.2200 & 0.2204 \end{bmatrix}, \; b = \begin{bmatrix} 0.4351 \\ -0.1929 \\ 0.6165 \\ -0.8022 \end{bmatrix}.$$

The solution x_0 of the equation $Ax = b$ was computed, and b was then perturbed to $b + \delta b$ where the elements of δb are drawn from a zero mean Gaussian distribution such that $\|b\| / \|\delta b\| = 6.39 \times 10^3$. The solution $\hat{x}_0 = x_0 + \delta x_0$ of the equation $Ax = b + \delta b$ was computed. The two solutions are

$$x_0 = \begin{bmatrix} 1.000075414240576 \\ -0.5000879795933287 \\ -0.02425113887960606 \\ 0.02624513954981467 \end{bmatrix}, \qquad \hat{x}_0 = \begin{bmatrix} 1.000133044180548 \\ -0.4999705917859966 \\ 0.08574824910454026 \\ -0.08372428931414788 \end{bmatrix}.$$

The relative errors in the components $x_{0,k}, k = 1, \ldots, 4$, of x_0 are

$$\begin{aligned} &|\delta x_{0,1}| / |x_{0,1}| = 5.76 \times 10^{-5}, \quad &&|\delta x_{0,2}| / |x_{0,2}| = 2.35 \times 10^{-4}, \\ &|\delta x_{0,3}| / |x_{0,3}| = 4.54, &&|\delta x_{0,4}| / |x_{0,4}| = 4.19, \end{aligned}$$

and the componentwise condition numbers are therefore, approximately,

$$\begin{aligned} &\frac{1}{\Delta b} \begin{bmatrix} |\delta x_{0,1}| / |x_{0,1}| & |\delta x_{0,2}| / |x_{0,2}| & |\delta x_{0,3}| / |x_{0,3}| & |\delta x_{0,4}| / |x_{0,4}| \end{bmatrix} \\ &= \begin{bmatrix} 0.37 & 1.51 & 2.91 \times 10^4 & 2.69 \times 10^4 \end{bmatrix}, \end{aligned}$$

where $\Delta b = \|\delta b\| / \|b\| = 1.56 \times 10^{-4}$. It is seen that $x_{0,1}$ and $x_{0,2}$ are well-conditioned, but $x_{0,3}$ and $x_{0,4}$ are ill-conditioned. The effective condition number of the equation is $S_{LS}(A, b) = 1.44 \times 10^3$ and the condition number of A is $\kappa(A) = 2.03 \times 10^3$. It is clear that the large value of $S_{LS}(A, b)$ is a reflection of the ill-conditioned nature of $x_{0,3}$ and $x_{0,4}$, but it fails to consider the stability of $x_{0,1}$ and $x_{0,2}$. □

This example shows that the effective condition number does not reveal how an error in b is distributed among the components of x_0. This information can be obtained by assigning a condition number to each component of x_0, and the next section considers these condition numbers.

2.3 Componentwise Condition Numbers

The condition numbers $\kappa_{LS}(A,\theta)$ and $S_{LS}(A,b)$ measure the errors in the normwise manner, but more refined estimates of the numerical condition of x_0 are obtained by assigning a condition number to each of its components. These condition numbers are introduced in [1] and expressions for them for the LS problem are derived. Since they consider the interaction of A and b, they differ from the collinearity measures that are introduced by Stewart [3], which only consider the linear dependence of the columns of A.

The componentwise condition number $R_{LS}(t_q,b)$ of the q'th component $x_{0,q}$ of x_0 is defined as the maximum value of the ratio of the relative error in $x_{0,q}$ to the relative error in b,

$$R_{LS}(t_q,b) := \max_{\delta b\in\mathbb{R}^m} \frac{\Delta x_{0,q}^{(c)}}{\Delta b}, \qquad \Delta x_{0,q}^{(c)} = \frac{|\delta x_{0,q}|}{|x_{0,q}|}.$$

Since $x_0 = A^\dagger b$, it follows that

$$x_{0,q} = t_q^T b,$$

where $t_q^T \in \mathbb{R}^m, q = 1,\ldots,n$, is the q'th row of $A^\dagger$, and thus

$$\frac{|\delta x_{0,q}|}{|x_{0,q}|} \le \frac{\|t_q\|\,\|\delta b\|}{|t_q^T b|},$$

and hence

$$R_{LS}(t_q,b) = \frac{\|t_q\|\,\|b\|}{|x_{0,q}|} = \frac{\|t_q\|\,\|b\|}{|t_q^T b|} = \frac{1}{|\cos\alpha_q|}, \qquad q = 1,\ldots,n, \tag{7}$$

where α_q is the angle between t_q and b.

The n condition numbers (7) are very refined because they measure the relative error in each component of x_0. They discriminate between the well-conditioned, and ill-conditioned, components of x_0, and they therefore provide information that cannot be deduced from the effective condition number.

The next section considers mixed condition numbers, which, as their name implies, measure the errors in the componentwise and normwise senses. They may therefore be considered as intermediate between the effective condition number and the componentwise condition numbers.

2.4 Mixed Condition Numbers

The mixed condition numbers that are considered in this section use both componentwise and normwise measures, and they therefore display features of the condition numbers that are considered in Sections 2.2 and 2.3.

The mixed condition numbers of the solution of the LS problem are defined as

$$T_{LS}\,(t_q, b) := \max_{\delta b \in \mathbb{R}^m} \frac{\Delta x_{0,q}^{(m)}}{\Delta b}, \qquad \Delta x_{0,q}^{(m)} = \frac{|\delta x_{0,q}|}{\|x_0\|},$$

and their derivation follows closely that of the componentwise condition numbers (7),

$$T_{LS}\,(t_q, b) = \frac{\|t_q\|\,\|b\|}{\|x_0\|} = \frac{\|t_q\|\,\|b\|}{\|S^{\dagger}U^T b\|}, \qquad q = 1, \ldots, n. \tag{8}$$

Example 4. Figure 5 shows the effective, componentwise and mixed condition numbers of the solution x_0 of the LS problem in Example 1 for $b = b_1$. It is seen that the componentwise condition numbers display a large variation in magnitude, and that some of the values are smaller, and some are larger, than the effective condition number. The mixed condition numbers are smaller than their equivalent componentwise condition numbers, and they are also smaller than the effective condition number. □

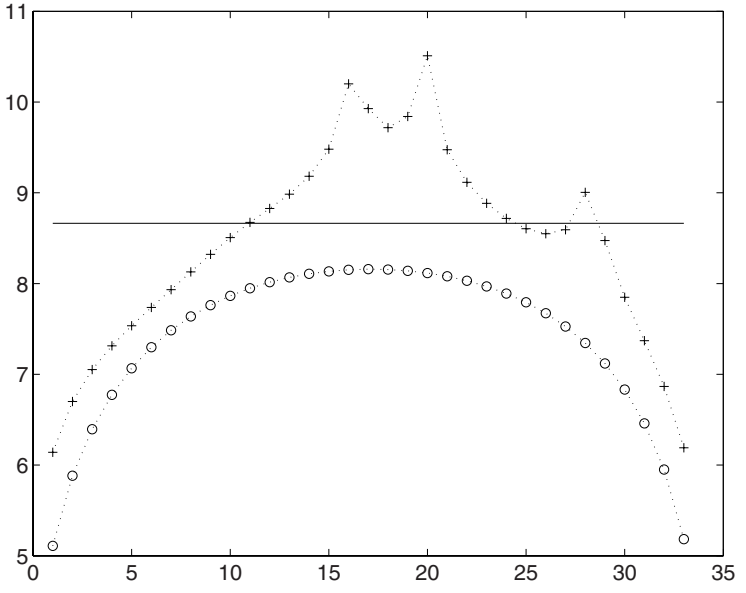


Fig. 5. The effective condition number (-), componentwise condition numbers (+) and mixed condition numbers (o) for the LS problem in Example 1 for $b = b_1$. A logarithmic scale is used for the condition numbers.

Figure 5 illustrates the general phenomenon that different condition numbers may differ by several orders of magnitude, and in particular, one or more components of x_0 may be well-conditioned, but x_0 may be ill-conditioned in the normwise sense. The next section establishes inequalities between the condition numbers in order to quantify the differences between the componentwise and normwise condition numbers.

3 Inequalities Between the Condition Numbers

The condition numbers that are discussed in Section 2 quantify the numerical stability of x_0, and the inequalities that unite them are considered in this section. The relationship between the effective condition number and $\kappa_{LS}(A, \theta)$ is established in (5), and this section considers the relationships between these condition numbers and the other condition numbers.

Consider initially the relationships between (i) the componentwise condition numbers and the effective condition number, and (ii) the mixed condition numbers and the effective condition number, for some, but not all, of the components of x_0. These are developed by noting that if $p \in \mathbb{R}^n$ is an arbitrary vector that satisfies $\|p\| = 1$, then the magnitude of at least one of its components must be greater than or equal to $1/\sqrt{n}$.

Since t_q^T is the q'th row of $A^\dagger$, it follows that

$$t_q^T = e_q^T A^\dagger = e_q^T V S^\dagger U^T, \tag{9}$$

where e_q is the q'th standard basis vector, and thus

$$\|t_q\| = \left\|e_q^T V S^\dagger\right\| \geq \left|e_q^T V S^\dagger e_n\right| = \left|v_q^T S^\dagger e_n\right| = \frac{|v_{q,n}|}{\sigma_n},$$

where v_q^T is the q'th row of V and $v_{q,n}$ is element (q, n) of V. Since at least one of the elements of the n'th column of V must be greater than or equal to $1/\sqrt{n}$, it follows that there exists at least one integer k, $1 \leq k \leq n$, such that

$$\|t_k\| \geq \frac{1}{\sqrt{n}\sigma_n}.$$

It therefore follows from (7) that there exists at least one integer k for which

$$R_{LS}(t_k, b) = \frac{\|t_k\| \, \|b\|}{|x_{0,k}|} \geq \frac{1}{\sqrt{n}\sigma_n} \frac{\|b\|}{|x_{0,k}|} = \frac{S_{LS}(A, b)}{\sqrt{n}} \frac{\|x_0\|}{|x_{0,k}|}. \tag{10}$$

This inequality shows that there are two contributory factors, a large value of the effective condition number and the condition $\|x_0\| \gg |x_{0,k}|$, to a large value of a componentwise condition number. Similarly, it follows from (8) that

$$T_{LS}(t_k, b) = \frac{\|t_k\| \, \|b\|}{\|x_0\|} \geq \frac{1}{\sqrt{n}\sigma_n} \frac{\|b\|}{\|x_0\|} = \frac{S_{LS}(A, b)}{\sqrt{n}}. \tag{11}$$

The inequalities (10) and (11) show that there exists at least one component of x_0 whose componentwise and mixed condition numbers, respectively, are large if the effective condition number is large. This condition is shown in Figure 5 for Example 1 with $b = b_1$.

The inequalities (10) and (11) may suggest that these condition numbers are consistent, but the following example shows that this desired consistency is not always present.

Example 5. Consider the equation

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a \\ \epsilon \end{bmatrix}, \qquad |a| \gg |\epsilon| > 0,$$

for which

$$S_{LS}(A, b) = \kappa(A) = 1 \qquad \text{and} \qquad T_{LS}\,(t_1, b) = T_{LS}\,(t_2, b) = 1,$$

and

$$R_{LS}\,(t_1, b) = \sqrt{1 + \left(\frac{\epsilon}{a}\right)^2} \approx 1 \qquad \text{and} \qquad R_{LS}\,(t_2, b) = \sqrt{1 + \left(\frac{a}{\epsilon}\right)^2} \gg 1.$$

It follows that the solution is well-conditioned in the normwise and mixed senses, but x_2 is ill-conditioned in the componentwise sense. □

The inequalities (10) and (11) define lower bounds for at least one, but not all, of the componentwise and mixed condition numbers of x_0, respectively. Lower bounds of these condition numbers that are satisfied by all the components of x_0 are now developed. Specifically, it follows from (9) that

$$\begin{aligned} \|t_q\| &= \left\|e_q^T V S^\dagger\right\| \\ &= \left\|v_q^T S^\dagger\right\| \\ &= \sqrt{\frac{v_{q,1}^2}{\sigma_1^2} + \frac{v_{q,2}^2}{\sigma_2^2} + \cdots + \frac{v_{q,n}^2}{\sigma_n^2}} \\ &= \frac{1}{\sigma_1}\sqrt{v_{q,1}^2 + v_{q,2}^2\left(\frac{\sigma_1}{\sigma_2}\right)^2 + \cdots + v_{q,n}^2\left(\frac{\sigma_1}{\sigma_n}\right)^2} \\ &\geq \frac{1}{\sigma_1}, \qquad q = 1, \ldots, n, \end{aligned}$$

and thus since

$$R_{LS}\,(t_q, b) = \frac{\|t_q\|\,\|b\|}{\left|t_q^T b\right|} = \frac{\|t_q\|\,\|b\|}{|x_{0,q}|} \geq \frac{\|t_q\|\,\|b\|}{\|x_0\|} = T_{LS}\,(t_q, b)\,,$$

it follows that

$$R_{LS}\,(t_q, b) \geq T_{LS}\,(t_q, b) = \frac{\|t_q\|\,\|b\|}{\|x_0\|} \geq \frac{1}{\sigma_1}\frac{\|b\|}{\|x_0\|}, \qquad q = 1, \ldots, n.$$

Equation (4) therefore yields

$$R_{LS}\left(t_q, b\right) \geq T_{LS}\left(t_q, b\right) \geq \frac{S_{LS}(A, b)}{\kappa(A)}, \qquad q = 1, \ldots, n, \tag{12}$$

which establishes the inequalities between the effective, componentwise and mixed condition numbers, and $\kappa(A)$. The first inequality states that the q'th componentwise condition number of x_0 is greater than or equal to the q'th mixed condition number of x_0, which is shown in Figure 5 for the LS problem in Example 1 for $b = b_1$.

Lower and upper bounds of the effective condition number are given in (5), and these bounds for the mixed condition numbers are now considered. In particular, it follows from (9) that

$$\|t_q\| \leq \frac{1}{\sigma_n}, \qquad q = 1, \ldots, n,$$

and thus the effective condition number is an upper bound for the mixed condition numbers,

$$T_{LS}\left(t_q, b\right) = \frac{\|t_q\| \, \|b\|}{\|x_0\|} \leq S_{LS}(A, b), \qquad q = 1, \ldots, n. \tag{13}$$

This equation can be combined with (5) and (12) to establish lower and upper bounds for the mixed condition numbers,

$$\frac{S_{LS}(A, b)}{\kappa(A)} \leq T_{LS}\left(t_q, b\right) \leq S_{LS}(A, b) \leq \kappa_{LS}(A, \theta), \qquad q = 1, \ldots, n, \tag{14}$$

and thus these condition numbers of x_0 may span several orders of magnitude if $\kappa(A) \gg 1$. For example, the mixed condition numbers that are shown in Figure 5 span three orders of magnitude.

The inequalities (14) contain condition numbers that are defined wholly in the normwise sense, and the componentwise condition numbers are not present. It is clear that a finite upper bound for the componentwise condition numbers does not exist because one or more of them may be infinite, and thus only lower bounds, which are given in (10) and (12), can be established for them.

The inequality (11) can be combined with (14) to show that there exists at least one integer k such that

$$\frac{S_{LS}(A, b)}{\sqrt{n}} \leq T_{LS}\left(t_k, b\right) \leq S_{LS}(A, b),$$

which restricts the mixed condition number of $x_{0,k}$ to a relatively small range.

The inequalities that have been obtained thus far consider the mixed and componentwise condition numbers of individual components of x_0. It is now shown that they are constrained, such that the n components of each of these condition numbers are not independent.

Consider initially the mixed condition numbers. Since

$$\begin{aligned}\sum_{q=1}^{n} T_{LS}^2\,(t_q, b) &= \frac{\|b\|^2}{\|x_0\|^2} \sum_{q=1}^{n} \|t_q\|^2 \\ &= \frac{\|b\|^2}{\|x_0\|^2} \left\|A^{\dagger}\right\|_F^2 \\ &= \frac{\|b\|^2}{\|x_0\|^2} \sum_{k=1}^{n} \frac{1}{\sigma_k^2} \\ &= S_{LS}^2\,(A, b) \sum_{k=1}^{n} \left(\frac{\sigma_n}{\sigma_k}\right)^2,\end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, it follows that the sum of the squares of the mixed condition numbers is constrained. Furthermore, since

$$1 \leq \sum_{k=1}^{n} \left(\frac{\sigma_n}{\sigma_k}\right)^2 \leq n,$$

it follows that

$$S_{LS}(A, b) \leq \sqrt{\sum_{q=1}^{n} T_{LS}^2\,(t_q, b)} \leq \sqrt{n}\, S_{LS}(A, b),$$

which is a tight restriction on the square root of the sum of the squares of the mixed condition numbers. It is readily verified that these bounds are consistent with (11) and (13), and they therefore confirm that if the effective condition number is large, there must be at least one component of x_0 whose mixed condition number is large.

This analysis can be repeated for the componentwise condition numbers,

$$\sqrt{\sum_{q=1}^{n} R_{LS}^2\,(t_q, b)} \geq S_{LS}(A, b).$$

4 Summary

Four condition numbers of the LS problem have been considered theoretically and computationally, and it has been shown that the simplest measure, $\kappa_{LS}(A, \theta)$, may overestimate, possibly by several orders of magnitude, the exact numerical condition of the LS problem. It follows that a numerically stable LS problem may have a large value of $\kappa(A)$, assuming that $\cos\theta = 1$, and that this may lead to incorrect conclusions, for example, that regularisation is required, even if its inclusion leads to significant errors in the computed solution. This disadvantage, which arises because this condition number is defined

as the maximum value of the error magnification of x_0 over all vectors b and δb, is overcome by the effective, mixed and componentwise condition numbers because they provide more precise information on the exact numerical condition of the LS problem. It was explained that these refined condition numbers are important in some applications of geometric modelling. An example of regression using radial basis functions was used to illustrate the differences between the condition numbers.

References

1. Chandrasekaran, S., Ipsen, I.C.F.: On the sensitivity of solution components in linear systems of equations. SIAM J. Mat. Anal. Appl. 16(1), 93–112 (1995)
2. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion. SIAM, Philadelphia, USA (1998)
3. Stewart, G.W.: Collinearity and least squares regression. Statistical Science 2(1), 68–100 (1987)
4. Watkins, D.S.: Fundamentals of matrix computations. John Wiley and Sons, New York, USA (1991)
5. Winkler, J.R.: A statistical analyis of the numerical condition of multiple roots of polynomials. Computers and Mathematics with Applications 45, 9–24 (2003)

Propagation of Geometric Tolerance Zones in 3D

Song-Hai Zhang[1], Qi-Hui Zhu[2], and Ralph R. Martin[3]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
zhangsh@cg.cs.tsinghua.edu.cn
[2] Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA
qihuizhu@seas.upenn.edu
[3] School of Computer Science, Cardiff University, Wales, UK
ralph@cs.cf.ac.uk

Abstract. This paper considers error propagation in three dimensional geometric constructions using a geometric approach. First, we present definitions and constructions of tolerance zones for various fundamental elements in Euclidean space. Then, we study in detail the propagation of errors during several geometric computations, including the the distance between two skew lines, reflections, projections, and rotations, and we derive new tolerance zones from the old ones.

1 Introduction

Computer-aided design and other geometric application areas create a demand for efficient and robust algorithms, which must often deal with imprecisely defined data. There is a need for reliable mathematical foundations of such algorithms, and in particular an understanding of how errors propagate during a chain of calculations with imprecise data.

A commonly used approach to this problem is to use interval arithmetic, which represents imprecise floating point numbers by intervals; some of this work is specific to geometric applications [1,3,2,7]. However, in general, interval methods tend to be too conservative, and overestimate the errors produced [6].

An alternative approach to dealing with errors in geometric computations is to use geometric tolerances as proposed by Wallner [9]. In this case, geometric tolerance zones are used, giving a more precise estimate of errors. They also have the important property of geometric invariance under rotations and translations, which interval methods based on coordinates do not [9]. Geometric tolerance computation and propagation have been studied in several papers, considering: errors in construction of geometric fundamental elements [6,9], tolerances of free-form curves [6,9], error propagation in geometric transformations [4], and geometric constraint solving [8]. However, much of this work has concentrated on two-dimensional problems.

R. Martin, M. Sabin, J. Winkler (Eds.): Mathematics of Surfaces 2007, LNCS 4647, pp. 494–507, 2007.
© Springer-Verlag Berlin Heidelberg 2007

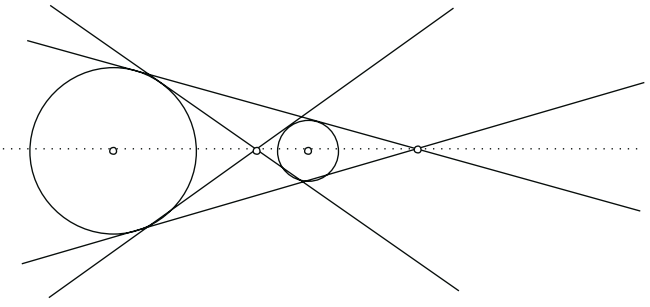


Fig. 1. Tolerance zone of line defined by two points

In this paper we consider tolerance zone computation and error propagation during various geometric computations in three dimensions, including the distance between two skew lines, and constructions including reflections, projections, and rotations.

2 Tolerances of Fundamental Geometric Elements

We start by defining imprecise geometric elements in three dimensions, which are the fundamental items used later in the paper. The simplest assumption to make is that the uncertainty is isotropic, i.e. an imprecisely known point lies somewhere in a sphere centred at its notional position. While this is the simplest model for a tolerance zone, other models may also be appropriate. For example, the uncertainty may be greater in a particular direction, or we may have separate ranges of uncertainty for each axis, leading to initial tolerance zones which are ellipsoidal, or box shaped, for example. In the following, we initially refer to a general 3D tolerance zone T for a point P, although later at times we assume more specifically that T is spherical, in order to provide simple results.

In many cases T may naturally be convex: for example, a sphere, ellipsoid, or polyhedron. Minkowski sums are useful when manipulating tolerance zones, as we will explain later. We note that the Minkowski sum of two convex objects is also a convex object—Minkowski sums can be computed much more easily and efficiently for convex objects than for general objects. If a tolerance zone is not naturally convex, it is often convenient to replace it by its convex hull, or some other simple convex shape, for this reason.

We start by defining the basic element, a point associated with a tolerance zone:

Definition 1. *A **fat point** is a 3D point with an associated tolerance zone T. If P is the notional position of the point, the fat point is the volume of space given by:*

$$FP(P) = \{Q | Q - P \in T\}.$$

We next define a fat line constructed from two fat points:

Definition 2. *A **fat line** is constructed from two fat points based on P_1, P_2, and is given by*

$$FL(P_1, P_2) = \{l(p_1, p_2) | p_1 \in FP(P_1), p_2 \in FP(P_2)\}$$

where $l(p_1, p_2)$ means the line through points p_1, p_2.

$FL(P_1, P_2)$ denotes the region covered by all lines having one point in $FP(P_1)$ and another point in $FP(P_2)$. Taking a specific case, if $FP(P_1)$ and $FP(P_2)$ are spherical tolerance zones, the resulting volume is bounded, in part, by two cones as shown in Fig. 1; one of the cones may degenerate into a cylinder.

We next define a fat plane constructed from three fat points:

Definition 3. *A **fat plane** is constructed from three fat points based on P_1, P_2, P_3 and is given by*

$$FPl(P_1, P_2, P_3) = \{\alpha(p_1, p_2, p_3) | p_1 \in FP(P_1), p_2 \in FP(P_2), p_3 \in FP(P_3)\}$$

where $\alpha(p_1, p_2, p_3)$ is the plane determined by three points.

When $FP(P_1)$, $FP(P_2)$, $FP(P_3)$ are all convex zones, $FPl(P_1, P_2, P_3)$ can be bounded by eight planes [6], each tangent to the tolerance zone for all three points.

Finally, we define a fat sphere with an associated tolerance zone, based on a fat point, and an uncertain radius. Let $S(P, r)$ be the sphere with centre P and radius r. Let $S^-(P, r)$ to denote its interior. Then:

Definition 4. *A **fat sphere** defined using the fat point based on P, with a radius range of $[r, R]$, is given by*

$$FS(P, r, R) = \bigcup_{p \in FP(P), l \in [r,R]} S(p, l)$$

If r is large enough, and, for instance, exceeds the radius of $FP(P)$, $FS(P, r, R)$ is a hollow object. Here the radius of $FP(P)$ is defined to be half of the maximum distance between any two points on the boundary. More precisely, $FS = FS_R - FS_r$. In many cases, for simplicity of calculation, a convex polyhedron will be used as the tolerance zone for a fat point. In such a case, we may write

$$FS_R = \bigcup_{p \in FP(P)} S^-(p, R), \qquad FS_r = \bigcap_{p \in FP(P)} S^-(p, r)$$

Here, FS_R is clearly the Minkowski sum of $S^-(0, R)$ and $FP(P)$. The expression for FS_r can be simplified to

$$FS_r = \bigcap_{p \in V(P)} S^-(p, r),$$

where $V(P)$ denotes the set of vertices of $FP(P)$. FS_r could be an empty set, if r is small. We now formally state this as a theorem for the case where the tolerance zone of the fat point is a convex polyhedron, and prove it:

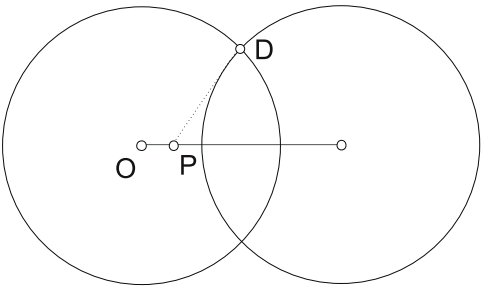


Fig. 2. Illustration of Lemma 1

Theorem 1. *Given a fat sphere defined as in Definition 4, when r is larger than the diameter of $FP(P)$, $FS(P, r, R)$ can be expressed as $FS(P, r, R) = FS_R - FS_r$, where*

$$FS_r = \bigcap_{p \in V(P)} S^-(p, r).$$

.

Proof. It can be readily seen that

$$\bigcap_{p \in V(P)} S^-(p, r) \supseteq \bigcap_{p \in P} S^-(p, r) = FS_r.$$

Thus, we have to prove the fact that $\forall p \in P$,

$$\bigcap_{q \in V(P)} S^-(q, r) \subseteq S^-(p, r)$$

and hence the validity of

$$\bigcap_{p \in V(P)} S^-(p, r) \subseteq \bigcap_{p \in P} S^-(p, r) = FS_r.$$

We do so using the following three lemmas:

Lemma 1. *In two dimensions, let $C^-(P, r)$ denote the interior of the circle centred at P with radius r. Any point P on the line segment between any two points P_1 and P_2 must satisfy $C^-(P, r) \supseteq C^-(P_1, r) \cap C^-(P_2, r)$.*

Proof. If $C^-(P_1, r) \cap C^-(P_2, r) = \emptyset$, then the lemma is obviously correct. Otherwise, we construct a circle centre O with radius r at either end of the line segment (O is P_1 or P_2). See Fig. 2. P is an arbitrary point on the line segment. Assume P is in the interior of the circle with center P_1, we can easily see that the intersection of the two circles centered at P_1 and P_2 respectively lies within the circle centered at P: distance PD is smaller than distance OD, i.e. $PD < r$. By symmetry, the same holds if P is in the interior of the circle with centre P_2. □

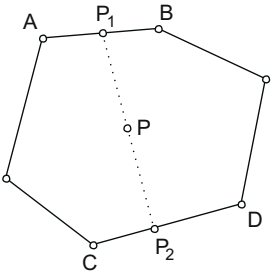


Fig. 3. Illustration of Lemma 3

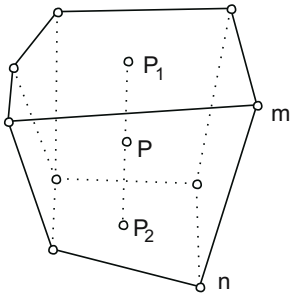


Fig. 4. Convex polyhedron

Lemma 2. *Generalising Lemma 1 to three dimensions for a sphere,* $S^-(P,r) \supseteq S^-(P_1,r) \cap S^-(P_2,r)$.

Lemma 3. *Let* $V(M)$ *be the vertex set of a polygon* M*. Given a point* P *inside the polygon* M*, Then*

$$S^-(P,r) \supseteq \bigcap_{Q \in V(M)} S^-(Q,r).$$

Proof. Construct some line through P that meets the boundary of the polygon at points P_1, P_2, on edges AB and CD, as shown in Fig. 3. According to Lemma 2,

$$S^-(P,r) \supseteq S^-(P_1,r) \cap S^-(P_2,r)$$

and

$$S^-(P_1,r) \supseteq S^-(A,r) \cap S^-(B,r), \quad S^-(P_2,r) \supseteq S^-(B,r) \cap S^-(C,r),$$

so

$$S^-(P,r) \supseteq S^-(A,r) \cap S^-(B,r) \cap S^-(C,r) \cap S^-(D,r) \supseteq \bigcap_{Q \in V(M)} S^-(Q,r). \quad \square$$

We now complete the proof of Theorem 1. For a point inside the polyhedron representing the tolerance zone of the fat point, construct some line intersecting

the faces of the polyhedron at P_1, P_2, as shown in Fig. 4. Using Lemmas 2 and 3, the following statements may now be proved in turn:

$$S^-(P,r) \supseteq S^-(P_1,r) \cap S^-(P_2,r),$$

$$S^-(P_1,r) \cap S^-(P_2,r) \supseteq \bigcap_{p \in V(m)} S^-(p,r) \cap \bigcap_{p \in V(n)} S^-(p,r),$$

$$\bigcap_{p \in V(m)} S^-(p,r) \cap \bigcap_{p \in V(n)} S^-(p,r) \supseteq \bigcap_{p \in V(M)} S^-(p,r),$$

as required. □

We now give an alternative definition of a fat sphere based on four fat points.

Definition 5. *The **fat sphere** generated by four fat points based on P_1, P_2, P_3, P_4 is given by*

$$FS(P_1,P_2,P_3,P_4) =$$
$$\{\sigma(p_1,p_2,p_3,p_4) | p_1 \in FP(P_1), p_2 \in FP(P_2), p_3 \in FP(P_3), p_4 \in FP(P_4)\}$$

where $\sigma(p,q,r,s)$ is the function which constructs a sphere through four points p, q, r, s.

We may now state the following theorem, assuming that the fat points in this case have spherical tolerance zones:

Theorem 2. *$\partial FS(P_1,P_2,P_3,P_4) \subseteq \Sigma$ where Σ is a set comprising 16 spheres that are each tangent to $FP(P_i)$, $i = 1,\ldots,4$.*

Proof. Suppose P is an arbitrary point on the boundary of the fat sphere. Let Ω be a sphere passing through P, determined by 4 points A, B, C, D, one from each fat point zone respectively. We will prove that Ω is tangent to all 4 fat zones. Let $\Omega(A)$ be a sphere passing through A and intersecting the other three fat point zones. Perform a polar transformation with center at A, so all spheres passing through A transfrom into planes and spheres not passing through A remain spheres afterwards. After this transformation, $\mathcal{P}(FP(P_i))$ are still spheres. $\mathcal{P}(\Omega(A))$ is a plane passing through A. In particular, $\mathcal{P}(\Omega(A))$ is the fat plane passing through the three spheres which are the images of three fat points under polar transformation. After polar transformation, if a point is originally interior to a sphere, it remains interiority to the image of the sphere. Points P in $(\Omega(A))$ are boundary points of the fat sphere if under polar transformation, there exists a plane passing through P', the image of P, which is tangent to images of three spheres. Thus, $\Omega(A)$ is tangent to all the spheres. More precisely, the boundary of the fat sphere is composed of 16 common tangent spheres: each sphere has two types of tangency, so four spheres have 16 different tangent spheres. □

3 Tolerance Zones and Geometric Computations in 3D

We now consider various geometric computations in three dimensions when performed on objects defined with respect to tolerance zones.

3.1 Skew Line Detection and Distance Computation

This section considers the issue of deciding whether two fat lines intersect, and the distance between two skew lines, assuming they are based on points with spherical tolerance zones. We combine the detection and distance computation into one algorithm. We assume the two fat lines are skew lines; if their distance is negative, then the lines intersect.

A continuity argument shows that the distance between two skew fat lines may be represented by a *single* interval, i.e. it can take on any value between a minimum, and a maximum, value.

As explained earlier, a fat line determined by two fat points is bounded by two cones. We compute the distance interval by considering pairs of defining cones, one from each fat line.

Thus, all we must consider is the maximum and minimum distance between two cones. Let θ_1, θ_2 denote the semi-angles of the two cones. and let h be the skew distance between their axes (see Figure 5). O_1 and O_2 are two origins, and l_1 and l_2 are two axes. Suppose points P_1 and P_2 lie on the two cones respectively, and the distance between these two points is minimum (or maximum). These points must satisfy:

1. P_1 and P_2 lie on the cone surfaces;
2. the line P_1P_2 is normal to the cones at P_1 and P_2, so it also passes through axis of each cone. Thus, the angle between line P_1P_2 and l_1 is $\pi/2 \pm \theta_1$, and the angle between line P_1P_2 and l_2 is $\pi/2 \pm \theta_2$.

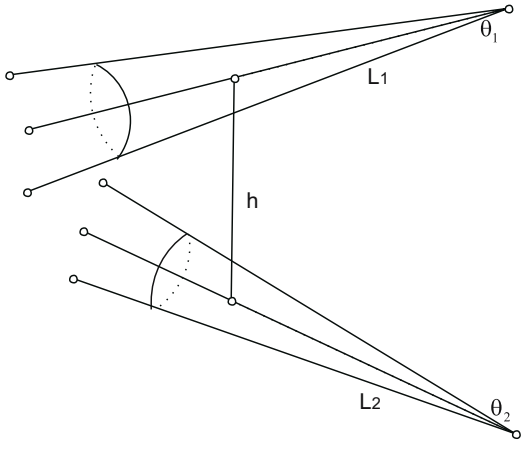


Fig. 5. Distance between two cones

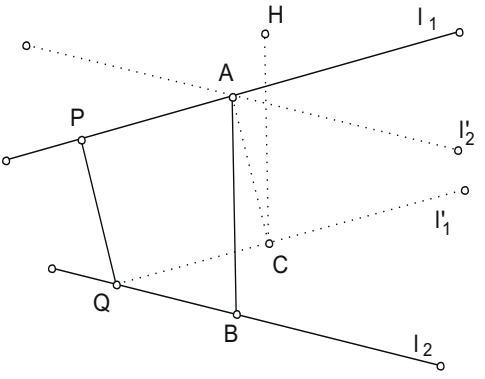


Fig. 6. Distance between two skew lines

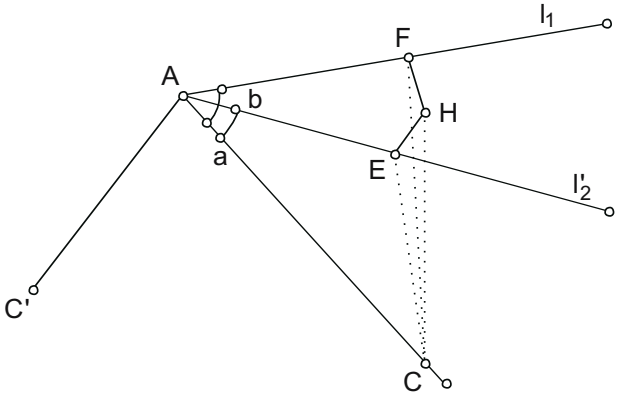


Fig. 7. Two possibilities for AC

Only lines *on* the cones need be considered to calculate maximum and minimum distances. Each fat line has two cones, so there are 4 combinations of cones to consider. We call the interval formed from the maximum and minimum distance arising from one combination a *partial distance* $[d_i^{\min}, d_i^{\max}]$. The continuity argument given before shows that the final result $[D_{\min}, D_{\max}]$ must be the union of these four intervals. The main problem is now to compute $d_i^{\min}$ and $d_i^{\max}$ for each pair of cones. This calculation can be done as a special case of the method given in [5], which considers the distance between a canal surface and a simple surface. Here we give an alternative, direct, approach.

See Figure 6. The axes of the cones under consideration are l_1, l_2, and their semi-angles are θ_1, θ_2. Points A and B are are the points of closest approach of the two axes (the distance between them is h). Points P and Q are the points where the line P_1P_2 meets each axis l_1, l_2 respectively. Note that the angles between PQ and l_1, l_2 respectively are $\frac{\pi}{2} - \theta_1$, $\frac{\pi}{2} - \theta_2$. Construct l_1', l_2' parallel to l_1, l_2, so that l_2' meets l_1 in A and l_1' meets l_2 in Q. Shift PQ to AC. AC is

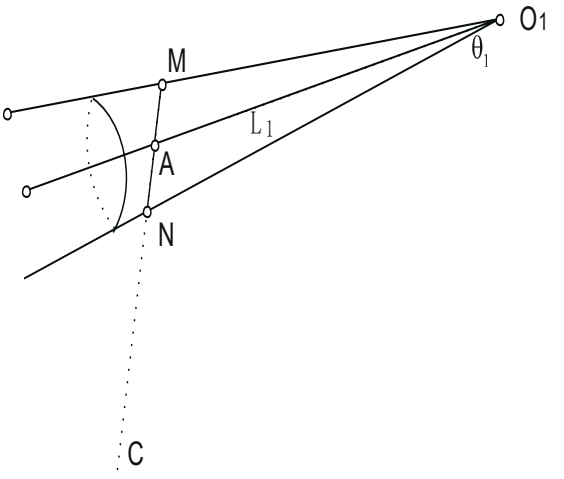


Fig. 8. The Distance inside cones

now either a minimum or maximum distance between two cones (equal to PQ). as desired. The distance CH, from C to the plane determined by l_1 and l_2', is exactly h, the extremal distance between the axes.

Figure 7 shows in further detail how to compute AC, and the extremal distance. The angles between AC and l_1, l_2 are $\pi/2 \pm \theta_1$, $\pi/2 \pm \theta_2$ respectively, giving four possible cases to consider for AC. Construct the projections E and F of H on l_1 and l_2' on the given plane. CF and FH, and CE and EH are both perpendicular to l_1, and l_2 respectively. The following formulae can be obtained, starting from $AC = D$, $CH = h$, and θ is the angle between l_1 and l_2':

$$AE = D\cos a, \quad AF = D\cos b, \quad EF = D\sqrt{\cos^2 a + \cos^2 b - 2\cos a\cos b\cos\theta},$$

$$AH = \frac{D\sqrt{\cos^2 a + \cos^2 b - 2\cos a\cos b\cos\theta}}{\sin\theta},$$

$$h = CH = \sqrt{D^2 - AH^2} = \frac{D\sqrt{\sin^2\theta - \cos^2 a - \cos^2 b + 2\cos a\cos b\cos\theta}}{\sin\theta},$$

$$D = \frac{h\sin\theta}{\sqrt{\sin^2\theta - \cos^2 a - \cos^2 b + 2\cos a\cos b\cos\theta}},$$

Substituting $a = \pi/2 \pm \theta_1$, $b = \pi/2 \pm \theta_2$, in which θ_1, θ_2 are acute angles, we get

$$D = \frac{h\sin\theta}{\sqrt{\sin^2\theta - \sin^2\theta_1 - \sin^2\theta_2 \pm 2\sin\theta_1\sin\theta_2\cos\theta}}.$$

In addition, the lengths of the line segments inside each cone must be added or subtracted from D to give the overall maximal or minimal distance (see Figure 8). These segments are AM and AN and their lengths should be added to

or subtracted from AC: note that AC is perpendicular to O_1M. We can easily find that

$$AM = O_1A/\sin\theta_1$$

$$AN = O_1A(\tan 2\theta_1/\cos\theta_1 - \sin\theta_1).$$

Determination of whether addition or subtraction is required depends on the relative positions of AC and the cone axis l_1.

The distance between two fat lines $[D_{\min}, D_{\max}]$ can thus be computed by combining four groups of minimum and maximum distances between cones.

3.2 Reflection and Line Projection in 2D

In this section we consider how to reflect an exact point in a fat line in 2D, using circular tolerance zones. A closely related problem is that of projecting an exact point onto a fat line in 2D: the projection of a point onto a line is the midpoint of the line joining the point and its reflection. Thus, a solution to either problem gives a solution to the other. Here, we consider 2D projection of an exact point onto a fat line; we then extend the ideas into 3D space in the next section.

If we wish to project a *fat* point onto a fat line, this can readily be done if the fat point is represented by a convex polygon. The tolerance zone of the projection can be found by computing the hull of the tolerance zones of the projection of each vertex of the polygon. The case in which the fat point is defined as a circular tolerance zone is more complicated, however, and further work is needed.

We start by giving a theorem concerning projection of an exact point P in a fat line.

Theorem 3. *Suppose we are given a 2D fat line $FL(O_1, O_2)$ defined in terms of two fat points with circular tolerance zones $FP(O_i)$, with O_i as center, and r_i as radius respectively. The boundary of the projection of an exact point P in this fat line can be expressed as a pair of curves $\rho = -l\cos\theta \pm r_i$, $i = 1, 2$ in polar coordinates, where the origin of polar coordinates is placed at P, and O_1P, O_2P are taken as $\theta = 0$ axes respectively.*

Proof. It can be easily seen that under projection, P goes to a region which is the intersection of two heart-shaped zones (H-zones for short), H_1, H_2 where H_i is the set of points H such that H is on the circumference of a circle with diameter PA, and $A \in FP(O_i)\}$. As Figure 9 shows, H is the projection point corresponding to some definite line joining some point A in $FP(O_1)$ to some point B in $FP(O_2)$: $PH \perp AH$ and $PH \perp BH$. Thus H belongs to both H_1 and H_2. We now consider how to calculate these two H-zones H_1, H_2.

See Figure 10, where O corresponds to either O_1 or O_2, i.e. the tolerance zone for either of the fat points. Construct a circle of diameter of PA, where A is an arbitrary point inside circle O. This circle intersects a line l passing through P at Q with $AQ \perp l$, i.e. Q is the projection of A on l. The H-zone consists of the envelope of all such circumferences. No point inside circle O can have a

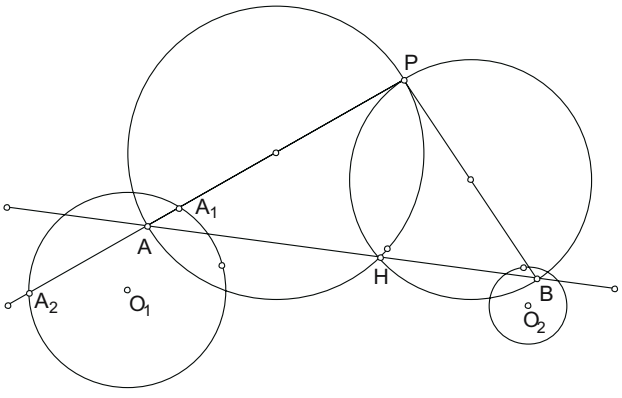


Fig. 9. Illustration of Theorem 3

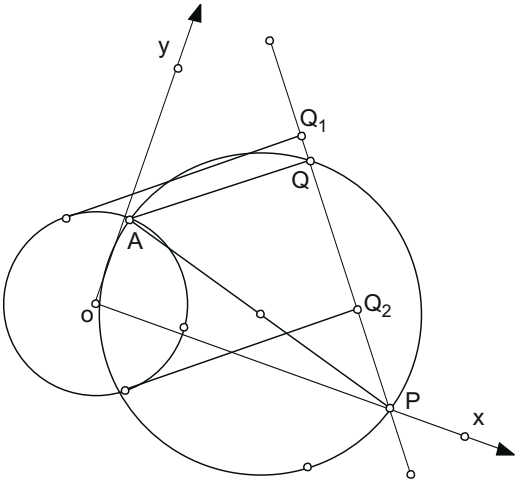


Fig. 10. Computation of inner and outer boundaries of H-zones

corresponding point Q which lies outside the line segment Q_1Q_2, bounded by the tangents to $FP(O)$ as shown in Figure 10. As l rotates through an angle ranging from $-\pi/2$ to $\pi/2$, the positions of Q_1 and Q_2 sweep out the inner and outer boundaries of the H-zone.

If l has an angle given by θ in polar coordinates with line PO, $\theta \in [-\pi, \pi]$. In polar coordinates r is the radius of circle O. Thus, we have

$$\overrightarrow{PQ_1} = -\overrightarrow{PO}\cos\theta + r,$$
$$\overrightarrow{PQ_2} = -\overrightarrow{PO}\cos\theta - r.$$

Hence the boundaries of the H-zone are $\rho = -\overrightarrow{PO_i}\cos\theta \pm r_i$, $i = 1, 2$. □

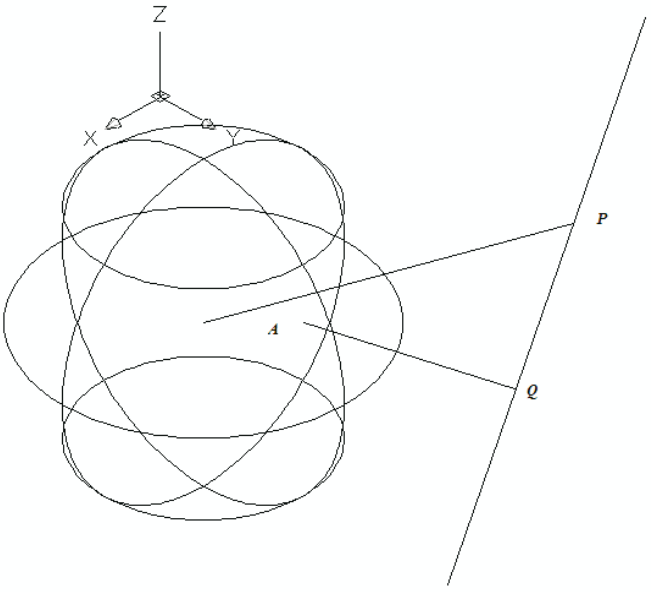


Fig. 11. Tolerance zone for 3D projection

Having found the projection of a point in a fat line, it is now straightforward to find the corresponding regions for *reflection* of a point in a fat line. The equations are simply:

$$\rho = -2\overrightarrow{PO_i}\cos\theta \pm 2r_i, \quad i = 1, 2.$$

3.3 Reflection and Plane Projection in 3D

We now consider projection and reflection of a point in a fat plane in 3D. Theorem 3 can be generalized as follows:

Theorem 4. *The tolerance zone formed by projection of a point P onto a fat plane in 3D is surrounded by 6 surfaces with equations in polar coordinates:*

$$\rho = l\cos\theta\cos\varphi \pm r_i, \quad i = 1, \ldots, 3$$

Proof. Following the ideas in 2D, the tolerance zone is the intersection of three H-zones H_1, H_2, H_3. To find its boundaries, consider an arbitrary line $l(\theta, \varphi)$, whose spatial orientation is given in spherical polar coordinates by θ, φ (eee Figure 11). The intersection of H_i and $l(\theta, \varphi)$ is a segment of l. Using the same notation as in the 2D case, if Q is on this segment, the following inequality holds:

$$l\cos\theta\cos\varphi - r \leq \overrightarrow{PQ} \leq l\cos\theta\cos\varphi + r$$

Thus, as the oriented line l takes on all orientations in 3D space, the boundaries of tolerance zone are, following the argument used in 2D:

$$\rho = 2l\cos\theta\cos\varphi \pm 2r \qquad \square$$

3.4 Rotation

Finally, we consider rotation of a fat point point $P(x, y, z)$ relative to an origin at Q, about an arbitrary axis. The transformation can be described by an orthogonal matrix R:

$$P' - Q = (P - Q)R$$

where P' is the position of P after rotation.

When there is uncertainty in Q, denoted by ΔQ, $P' - Q - \Delta Q = (P - Q - \Delta Q)R$ holds for all possible positions of Q. We may rearrange this as

$$P' - Q = (P - Q)R + \Delta Q - \Delta QR.$$

Consider the term arising due to uncertainty, $\Delta Q - \Delta QR$. If the rotation angle is θ, the magnitude of this error term is $2\sin\theta/2$, and it makes an angle $\pi/2 - \theta/2$ relative to ΔQ. Thus, $\Delta Q - \Delta QR$ is obtained by rotation through an angle $\pi/2 - \theta/2$ and stretching by a ratio of $2\sin\theta/2$.

If we now consider the specific case where the uncertainty in Q is a sphere with radius r, the uncertainty after rotation is still a sphere, but with a radius $2r\sin\theta/2$.

Let us now further suppose that there is an uncertainty ΔP in the original point P. The total uncertainty is now $\Delta PR + \Delta Q - \Delta QR$. This can be viewed as the Minkowski sum of the uncertainty in P after rotation, with $\Delta Q - \Delta QR$. If ΔP is also represented by a sphere independently of ΔQ, the final result is a larger sphere of radius $r_p + 2r\sin\theta/2$, where r_p is the size of the spherical tolerance zone of P.

4 Conclusion and Future Work

We have provided definitions and constructions for various three dimensional geometric elements with tolerance, including points, lines, planes and spheres. We have then shown how to perform several geometric computations which take these tolerances into account, and provide suitable tolerance zones for the output. These calculations include the distance between two skew lines, and various reflections, projections, and rotations, with spherical tolerance zones. We have also discussed other computations with polyhedral tolerance zones.

Representation of tolerance zone boundaries is an important issue, and still needs further work. Boundaries of tolerance zones need to be restricted to certain particular shapes (e.g. spheres and polyhedra) if algorithms are to be efficiently implemented.

References

1. Abrams, S.L., Cho, W., Hu, C.-Y., Maekawa, T., Patrikalakis, N.M., Sherbrooke, E.C., Ye, X.: Efficient and reliable methods for rounded-interval arithmetic. Computer-aided Design 30(8), 657–665 (1998)
2. Hu, C.-Y., Maekawa, T., Patrikalakis, N.M., Ye, X.: Robust interval algorithm for surface intersections. Computer-aided Design 29(9), 617–627 (1997)
3. Hu, C.-Y., Patrikalakis, N.M., Ye, X.: Robust interval solid modelling part I: representations. Computer-aided Design 28(10), 807–817 (1996)
4. Hu, S.-M., Wallner, J.: Error propagation through geometric transformations. Journal for Geometry and Graphics 2(8), 171–183 (2004)
5. Kim, K.-J.: Minimum distance between a canal surface and a simple surface. Computer-Aided Design 35(10), 871–879 (2003)
6. Pottmann, H., Odehnal, B., Peternell, M., Wallner, J., Haddou, R.A.: On optimal tolerancing in computer-aided design. In: Martin, R., Wang, W. (eds.) Proceedings of the Conference on Geometric Modeling and Processing (GMP-00), Los Alamitos, pp. 347–363. IEEE, Orlando (2000)
7. Tuohy, S.T., Maekawa, T., Shen, G., Patrikalakis, N.M.: Approximation of measured data with interval B-splines. Computer-Aided Design 29(11), 791–799 (1997)
8. Wallner, J., Schröcker, H.-P., Hu, S.-M.: Tolerances in geometric constraint problems. Reliable Computing 3(11), 235–251 (2005)
9. Wallner, J., Krasauskas, R., Pottmann, H.: Error propagation in geometric constructions. Computer-Aided Design 32(11), 631–641 (2000)

Author Index

Alberti, Lionel 1
Alliez, Pierre 29
Almegaard, Henrik 42
Augsdorfer, Ursula H. 391

Bagger, Anne 42
Benton, Phillips A. 64
Bez, Helmut E. 78
Bobach, Tom 93

Caravantes, Jorge 104
Cashman, Thomas J. 121, 391
Cripps, Robert J. 307

Davydov, Oleg 138
Dodgson, Neil A. 121, 391

Elber, Gershon 275

Farin, Gerald 93
Fujimori, Tomoyuki 442

Ginkel, Ingo 166
Gonzalez-Vega, Laureano 104
Gravesen, Jens 42
Gu, Xianfeng David 209

Hancock, Edwin R. 412, 427
Hansford, Dianne 93
Hung, Chao-Kuei 177

Inselberg, Alfred 177

Jin, Miao 209
Jüttler, Bert 42, 245

Karčiauskas, Kestutis 233
Kim, Junho 209
Kosinka, Jiří 245

Langbein, Frank C. 290
Langer, Torsten 263
Lev, Ronen 275
Liu, Shenglan 290
Liu, Yang 341

Ma, Xinhui 307
Martin, Ralph R. 290, 494
Michikawa, Takashi 442
Miwata, Yasuhiko 442
Mourrain, Bernard 1
Mullineux, Glen 378

Pérez-Díaz, Sonia 320
Peters, Jorg 233
Pottmann, Helmut 341

Reif, Ulrich 364
Robinson, Sebastian T. 378
Rosin, Paul L. 290

Sabin, Malcolm A. 121, 391
Sadaoka, Noriyuki 442
Saucan, Emil 275
Schumaker, Larry L. 138
Seidel, Hans-Peter 154, 263
Šír, Zbynek 42
Smith, William A.P. 412, 427
Suzuki, Hiromasa 442

Theisel, Holger 154

Umlauf, Georg 93, 166

von Funck, Wolfram 154

Warder, John P. 465
Winkler, Joab R. 480

Zhang, Song-Hai 494
Zhu, Qi-Hui 494