

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

New York University, NY, 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

Yong-Jin Kwon Alain Bouju
Christophe Claramunt (Eds.)

Web and Wireless Geographical Information Systems

4th International Workshop, W2GIS 2004
Goyang, Korea, November 2004
Revised Selected Papers



Springer

Volume Editors

Yong-Jin Kwon

Hankuk Aviation University

200-1, Hwajon-dong, Deokyang-gu, Goyang-city, Gyeonggi-do, 412-791, Korea

E-mail: yjkwon@tikwon.hangkong.ac.kr

Alain Bouju

University of La Rochelle

Faculty of Sciences and Technology, L3i Research Laboratory

Avenue Michel Crepeau, 17042 La Rochelle Cedex 1, France

E-mail: alain.bouju@univ-lr.fr

Christophe Claramunt

Naval Academy Research Institute

Lanveoc-Poulmic, BP 600, 29240 Brest Naval, France

E-mail: claramunt@ecole-navale.fr

Library of Congress Control Number: 2005925862

CR Subject Classification (1998): H.2, H.3, H.4, H.5, C.2

ISSN 0302-9743

ISBN-10 3-540-26004-8 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-26004-2 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

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik

Printed on acid-free paper SPIN: 11427865 06/3142 5 4 3 2 1 0

Preface

The aim of the annual W²GIS workshop is to provide an up-to-date review of advances on recent development of Web and wireless geographical information systems, and new challenges and opportunities for researchers, developers and users in the GIS community. The main topic of the W²GIS workshop is theoretical and technical issues of Web and wireless geographical information systems. This workshop followed the successful 2001, 2002 and 2003 editions, held in Kyoto, Singapore and Rome, respectively. The 2004 edition was held in Goyang, Korea.

In its 4th year, W²GIS reached new heights of recognition as a quality workshop for the dissemination and discussion of new ways of accessing and analyzing geospatial information. This year, 39 papers were submitted from 15 countries, and 20 papers were accepted from 11 countries. Similarly, the Program Committee consisted of 39 members from 16 countries.

We had the privilege of having three distinguished invited talks: “Eliciting User Preferences in Web Urban Spaces,” Yanwu Yang and Christophe Claramunt, Naval Academy Research Institute, France; “Discovering Regional Information from Web: Localness and Landmark Computation,” Katsumi Tanaka, Department of Social Informatics, Graduate School of Informatics, Kyoto University, Japan; and “Towards Knowing, Always and Everywhere, Where Everything Is, Precisely,” Christian S. Jensen, Department of Computer Science, Aalborg University, Denmark.

The workshop was organized by the Internet Information Retrieval Research Center (IRC) of Hankuk Aviation University, Korea. The IRC was established in 2001, sponsored by the Ministry of Science and Technology, Korea, KOSEF (the Korea Science and Engineering Foundation), Gyonggi-do (local government) and various venture companies (over 20). The research term of the IRC is 9 years, and its research fund is 9 million dollars total.

We wish to thank the authors for the high quality of their papers and presentations, and the Program Committee members for their timely and rigorous reviews of the papers. We thank the Steering Committee members for organizing and supporting this event. Finally, special thanks go to Stefano Spaccapietra for his advice and help.

November 2004

Yong-Jin Kwon
Alain Bouju
Christophe Claramunt
Workshop Chairs
W²GIS 2004

In Memory of Prof. Yahiko Kambayashi (February 15, 1943 – February 6, 2004)

We are deeply saddened by the sudden and untimely passing away of Prof. Yahiko Kambayashi, Dean of the School of Informatics, Kyoto University, Japan. His unexpected departure is a tremendous shock to everyone who knew him. He was 60.

Prof. Kambayashi was one of the pioneers of database research as well as a leader of the international database research community. He published numerous articles in major database journals and conferences such as Information Systems, SIGMOD, VLDB and ICDE. He was also the author and the editor of many books and conference proceedings. Prof. Kambayashi was an IEEE fellow, a trustee of the VLDB Endowment, a member of the SIGMOD Advisory Committee, a vice-chair of the ACM Tokyo/Japan Chapter, the chair of the DASFAA Steering Committee, a co-chair of the WISE Society and the WISE Steering Committee, a member of the W²GIS Steering Committee, a member of the CODAS Steering Committee, a member of the ER Steering Committee, a member of the RIDE Steering Committee, a co-editor-in-chief of the World Wide Web Journal, an associate editor of ACM TODS, and a member of the editorial board of several international journals. He was a winner of the ACM SIGMOD Contribution Award in 1995 for his many professional services in Japan and worldwide.

Prof. Kambayashi had been making great efforts to organize this W²GIS 2004 workshop, just before his sudden passing. Together with all the authors and the participants, we are taking this opportunity to express our heartfelt condolence and our deepest sympathy to his family.

November 26th, 2004

W²GIS 2004 Workshop Committee

W²GIS 2004 Workshop Committee

Honorary Chair

The late Yahiko Kambayashi (Kyoto University, Japan)

Workshop Chairs

Yong-Jin Kwon (Hankuk Aviation University, Korea)

Alain Bouju (University of La Rochelle, France)

Steering Committee

Michela Bertolotto (University College Dublin, Ireland)

Christophe Claramunt (Naval Academy, France)

Bo Huang (National University of Singapore, Singapore)

Hui Lin (Chinese University of Hong Kong, China)

Stefano Spaccapietra (EPFL, Switzerland)

Program Committee

Gennady Andrienko (GMD, Germany)

Masatoshi Arikawa (Tokyo University, Japan)

Chaitan Baru (San Diego Supercomputer Center, USA)

R.A. de By (ITC, The Netherlands)

James Carswell (Dublin Institute of Technology, Ireland)

Young-Sik Choi (Hankuk Aviation University)

Matt Duckham (NCGIA, USA)

Max Egenhofer (NCGIA, USA)

Mark Gahegan (Penn State, USA)

Ki-Joon Han (Konkuk University, Korea)

Mizuho Iwaihara (Kyoto University, Japan)

Bin Jiang (Gävle University, Sweden)

Myoung Ah Kang (University of Clermont-Ferrand, France)

Menno-Jan Kraak (ITC, The Netherlands)

Robert Laurini (INSA, France)

Keung-Hae Lee (Hankuk Aviation University, Korea)

Jong-Hoon Lee (ETRI, Korea)

Ki-Joune Li (Pusan National University, Korea)

Yoshifumi Masunaga (Ochanomizu University, Japan)

VIII Workshop Committee

Pedro Muro-Medrano (Universidad de Zaragoza, Spain)
Beng C. Ooi (National University of Singapore, Singapore)
Evtim Peytchev (Nottingham Trent University, UK)
Keun-Ho Ryu (Chungbuk National University, Korea)
Krithi Ramamrithan (Indian Institute of Tech., India)
Shashi Shekhar (Kyoto University, Japan)
Hiroki Takakura (Kyoto University, Japan)
Hiroya Tanaka (Tokyo University, Japan)
Katsumi Tanaka (Kyoto University, Japan)
George Taylor (University of Glamorgan, UK)
Remi Thibaud (Naval Academy, France)
Kristian Torp (University of Aalborg, Denmark)
Christelle Vangenot (EPFL, Switzerland)
Jari Veijalainen (GITI, Waseda University, Japan)
Kyu-Young Whang (KAIST, Korea)
Ilia Zaslavsky (San Diego Super Computer Center, USA)
Xiofang Zhou (University of Queensland, Australia)

Organization Committee

Soo-Hwan Chae (Vice-Director of the IRC, Korea)
Joong-Hwan Baek (Hankuk Aviation University, Korea)
Soo-Chan Hwang (Hankuk Aviation University, Korea)
Kang-Won Lee (Vice-President of HIST Co., Ltd., Korea)
In-Man Park (Vice-President of ILDO Eng., Co., Ltd., Korea)
Jong Sou Park (Hankuk Aviation University, Korea)
Joonseon Ahn (Hankuk Aviation University, Korea)
Yong-Ik Yoon (Sookmyung Women's University, Korea)
Syng-Yup Ohn (Hankuk Aviation University, Korea)
Yong-Hee Jang (Hankuk Aviation University, Korea)
Il-Young Moon (Hankuk Aviation University, Korea)

Sponsors

Sponsored by

Internet Information Retrieval Research Center (IRC), HAU, Korea
Ministry of Information and Communication, Korea
Korea Science and Engineering Foundation
Gyeonggi-do
Hanjin Information Systems & Telecommunication Co., Ltd.

Academic Sponsors

GIS Association of Korea
Korea Open Geographic Information Systems Association

In Cooperation with

Informatics Research Center for the Development of a Knowledge Society
Infrastructure, Kyoto University, Japan

Table of Contents

Web GIS

Web Services Framework for Geo-spatial Services	1
<i>Minsoo Kim, Mijeong Kim, Eunkyu Lee, and Inhak Joo</i>	
Temporal and Spatial Attribute Extraction from Web Documents and Time-Specific Regional Web Search System	14
<i>Taro Tezuka and Katsumi Tanaka</i>	

Mobile GIS and LBS

Broadcasting and Prefetching Schemes for Location Dependent Information Services	26
<i>KwangJin Park, MoonBae Song, and Chong-Sun Hwang</i>	
A Geocoding Method for Natural Route Descriptions Using Sidewalk Network Databases	38
<i>Kouzou Noaki and Masatoshi Arikawa</i>	
Location-Based Tour Guide System Using Mobile GIS and Web Crawling	51
<i>Jong-Woo Kim, Chang-Soo Kim, Arvind Gautam, and Yugyung Lee</i>	
A Progressive Reprocessing Transaction Model for Updating Spatial Data in Mobile Computing Environments	64
<i>Donghyun Kim and Bonghee Hong</i>	

Interoperability and Security in W²GIS

Mediation for Online Geoservices	81
<i>Omar Boucelma and François-Marie Colonna</i>	
A Generic Framework for GIS Applications	94
<i>Miguel R. Luaces, Nieves R. Brisaboa, José R. Paramá, and Jose R. Viqueira</i>	
Intrusion Detection System for Securing Geographical Information System Web Servers	110
<i>Jong Sou Park, Hong Tae Jin, and Dong Seong Kim</i>	

Indexing and Query Processing in W²GIS

In-route Skyline Querying for Location-Based Services	120
<i>Xuegang Huang and Christian S. Jensen</i>	
P2P Spatial Query Processing by Delaunay Triangulation	136
<i>Hye-Young Kang, Bog-Ja Lim, and Ki-Joune Li</i>	
Expansion-Based Algorithms for Finding Single Pair Shortest Path on Surface	151
<i>Ke Deng and Xiaofang Zhou</i>	
MR-Tree: A Cache-Conscious Main Memory Spatial Index Structure for Mobile GIS	167
<i>Kyung-Chang Kim and Suk-Woo Yun</i>	

Map Services for LBS

Developing Non-proprietary Personalized Maps for Web and Mobile Environments	181
<i>Julie Doyle, Qiang Han, Joe Weakliam, Michela Bertolotto, and David Wilson</i>	
Labeling Dense Maps for Location-Based Services	195
<i>Qing-Nian Zhang</i>	
Mobile SeoulSearch: A Web-Based Mobile Regional Information Retrieval System Utilizing Location Information	206
<i>Yong-Jin Kwon and Do-Hyoung Kim</i>	

3-D GIS and Telematics

A Novel Indexing Method for Digital Video Contents Using a 3-Dimensional City Map	221
<i>Yukiko Sato and Yoshifumi Masunaga</i>	
VRML-Based 3D Collaborative GIS: A Design Perspective	232
<i>Z. Chang and Songnian Li</i>	
Arrival Time Dependent Shortest Path by On-road Routing in Mobile Ad-Hoc Network	242
<i>Kyoung-Sook Kim, So-Young Hwang, and Ki-Joune Li</i>	
Author Index	255

Web Services Framework for Geo-spatial Services

Minsoo Kim, Mijeong Kim, Eunkyoo Lee, and Inhak Joo

Telematics Research Division, ETRI
161, Kajeong-dong, Yuseong-gu, Daejeon, South Korea
{minsoo, kmj63341, ekyulee, ihjoo}@etri.re.kr
<http://www.etri.re.kr/e-etri/>

Abstract. In this paper, we propose a Web Service framework for four kinds of geo-spatial services of GIS, SIIS, ITS, and GNSS. First, we examine what requirements are needed when designing the framework for various kinds of geo-spatial services. Then, we show how the framework can contribute to efficient geo-spatial services on both wired and wireless environment. Main issues in a design of the framework are to define interoperable interfaces, to define standardized metadata, and to design efficient geo-spatial server for various kinds of geo-spatial services. The framework fundamentally adopts international standards such as WMS, WFS, WCS, and WRS announced by OGC. The adoption satisfies the interoperability, extensibility, and standardization of the framework. Especially, we focus on a design of main memory-based GIS server(MM server). The MM server can efficiently serve huge volume of GML documents via Web Service. And experimental results show the effectiveness of the framework and MM server.

1 Introduction

Recent advance in web-based technology is raising new users' requirements that intend to serve complex and huge volume of information via Web. Especially, in geo-spatial information research field, such requirements that intend to serve huge volume of vector-typed map and satellite imagery map via both wired and wireless network are emerging. Over the past few years, a great deal of attention [1–6] in the Web Service technology has been directed towards efficient service integration among heterogeneous distributed servers. Therefore, it is meaningful to apply the Web Service technology to services integration among heterogeneous geo-spatial servers. However, it is not easy to provide geo-spatial services on Web environment on account of diversity and complexity [7] of geo-spatial data themselves. Actually, there were nearly case studies that can serve huge volume of geo-spatial data owing to the efficiency problem of Web Service so far.

So, in this paper we propose *Web Service framework* for geo-spatial services that has features of interoperability on Web environment, extensibility on various kinds of geo-spatial services, and efficient performance on wired and wireless network. The framework is composed of several kinds of *geo-spatial*

servers, *geo-spatial broker*, and *web-based clients*. In order to provide interoperability and extensibility on the geo-spatial services, we basically apply implementation specifications of OpenGIS Consortium (OGC) for designing the Web Service framework [8]. Speaking in detail, “*The Simple Features Specification for OLE/COM*”(SFS) [9], “*Web Map Services Implementation Specification*”(WMS) [10], “*Web Feature Services Implementation Specification*”(WFS) [11], “*Web Coverage Services Implementation Specification*” (WCS) [12], “*Geography Markup Language Implementation Specification*”(GML) [13], and “*Web Registry Services Implementation Specification*”(WRS) [14] are used to build the framework. The framework, actually, uses WMS, WFS, and WCS to build various kinds of geo-spatial servers that can provide image map such as JPG or GIF, vector map such as GML, and coverage information such as GeoTIFF, respectively. WRS is used to build geo-spatial broker that can provide a runtime discovery and registration for geo-spatial services. This Web Service framework can support four kinds of geo-spatial services that GIS server, SIIS (Spatial Imagery Information System) server, ITS server and GNSS server expose. Geo-spatial broker that accommodates WRS and references “*Universal Description, Discovery and Integration*” (UDDI) is a kind of metadata repository for geo-spatial services. So, geo-spatial servers can publish metadata for their services to the geo-spatial broker, and web-based clients can find metadata for their requests from the broker. Especially, the framework includes *MM server* that can give fast responses to clients by removing loading time of geo-spatial data and conversion time for vector data to GML. The main advantages of this Web Service framework are (1) it can efficiently provide the integrated services of various kinds of geo-spatial servers by using OGC and W3C specifications; (2) it can give fast responses to clients by using the MM server.

First, we present an overview of Web Service environment for geo-spatial services in the next section. Then, we present specific requirements of the Web Service framework for geo-spatial services in section 3 and propose our solutions that satisfy those requirements in section 4. Lastly, we show some examples for the framework and conclude our suggestions in the remaining sections.

2 Overview of Web Service Environments for Geo-spatial Services

A Web Service provides a set of protocols that allow applications to expose their functions and data to other applications over the Internet. Also, a Web Service provides a language and platform independent syntax for exchanging complex objects using messages. This Web service architecture has three essential components: *service provider*, *service broker*, and *service requestor*. Service provider publishes an availability of its service resources to service broker using “*Web Service Description Language*”(WSDL) [15] and delivers its services to service requestor when the service requestor wants to bind to its services. Service broker is acting as a registry or clearinghouse of services using UDDI. Service requestor performs discovery operations of services from the service broker and receives

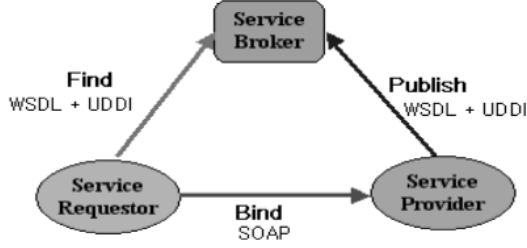


Fig. 1. Web Service Architecture.

result metadata. Using the metadata, service requestor binds to services of service provider and performs services. Such mechanism of Web Service is termed as “*Publish-Find-Bind*” and can be depicted as Fig. 1. Web environments for geo-spatial data are mostly researched and standardized by OGC. The OGC has announced many implementation specifications for Internet service of various kinds of geo-spatial data. However, the old specifications that have already announced by OGC considered only Internet services for geo-spatial data on web browser. In the beginning, OGC announced WMS, WFS, WCS, and Catalog services specification [16] to easily provide geo-spatial data on web browser. They did not consider the Web Service architecture that can expose geo-spatial services directly using XML. However, OGC began to cooperate with W3C recently, and OGC is announcing new specifications such as WRS, GML, and “*OpenGIS Reference Model*”(ORM) [17] for Web Service. Although the new specifications are not perfectly conformed to Web Service architecture, OGC are making all efforts to confirm the new specifications to Web Service specifications. Speaking in detail, WMS only defines interfaces that can transmit simple image map for web browser based on server-side mapping. On the contrary, WFS defines more complex interfaces than WMS, which can transmit GML documents and can perform spatial operators based on client-side mapping. WCS is similar to WFS except that it transmits satellite imagery data instead of vector-typed map. GeoTIFF, HDF-EOS, DTED, and NITF are widely used as encoding formats for the satellite imagery data. There are additional specifications of “*Styled Layer Descriptor*”(SLD) [18] and “*Coverage Portrayal Service*”(CPS), which define interfaces to change client-side mapping into server-side mapping in case of WFS and WCS servers. In other words, SLD and CPS can convert GML and GeoTIFF typed data that are generated in server side mapping into image map that can be used in client. Such conversions are needed in case of thin clients that can only visualize image map. Fig. 2 shows relations between OGC specifications that serve various kinds of geo-spatial data. In this paper, the Web Service framework adopts OGC specifications of WMS, WFS, WCS, and WRS as a basic architecture. Therefore, the framework can serve all kinds of geo-spatial data that OGC defines. Additionally, we upgrades and modifies the basic framework in order to efficiently serve huge volume of geo-spatial services on both wired and wireless network environment, which will be discussed in section 4.

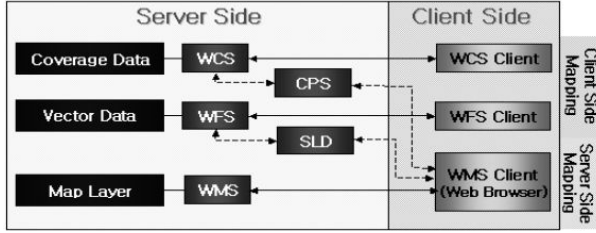


Fig. 2. Relation between OGC Specifications for Geo-spatial Data.

3 Requirements and Problems for Web Service Framework

In this section, we focus on the properties of various kinds of geo-spatial data and services that should be considered when we design the Web Service framework.

3.1 Huge Volume of Geo-spatial Data

Huge volume problem is an inherent property of geo-spatial data. Generally, the size of one geo-spatial object varies from tens of bytes to thousands of Kbytes in the real world applications. Because of huge volume of objects and large number of objects, it is almost impossible to efficiently transmit them to web clients on Web Service environment. General geo-spatial servers, for example Microsoft TerraServer [19], need three steps to provide huge volume of geo-spatial data using Web Service: (1) load geo-spatial objects from the persistent storage such as file system or database system; (2) make XML documents, GML document in this case, for loaded objects; and (3) transmit huge volume of GML documents. Most geo-spatial servers that mainly handle huge volume and large number of objects consume too much time in the three steps. We think it is very difficult to provide geo-spatial data using Web Service on wired and wireless network, so we propose MM server as a solution to solve the time constraint. The MM server is explained in detail in the next section.

3.2 GML-Typed Geo-spatial Data

For the purpose of satisfying Web Service architecture, it is also an important requirement for GIS server to convert original geo-spatial objects into GML documents on-line. Unfortunately the GML conversion consumes too much time, which often becomes major cause of late response to clients. Moreover, as the size of GML document increases, the response time to clients is getting worse and worse. So, we propose preprocessing of GML conversion as a solution for that problem. The preprocessing can give fast response to clients by saving the time consumed by on-line GML conversion. We will discuss more details about this in the next section.

3.3 Web Service for Geo-spatial Services

Web Service framework for geo-spatial services should provide not only geo-spatial data, but also geo-spatial functions. In other words, geo-spatial servers should be able to publish metadata for geo-spatial data and functions, clients should be able to find out metadata of the data and functions from broker, and clients should be able to bind to the data and functions. So, clients can access geo-spatial data directly or indirectly using geo-spatial functions. However, the old specifications of OGC, especially WRS, basically did not consider the Web Service architecture. Therefore, we propose a new geo-spatial broker as an extended model of WRS by adding UDDI specification. More detailed description is shown in the next section.

4 Solution: Web Service Framework for Geo-spatial Services

This section presents the Web Service framework to meet the requirements and to solve the problems as stated above. The framework consists of four kinds of *geo-spatial servers*, a *geo-spatial broker*, and *web-based clients*. Comparing Web Service architecture, geo-spatial servers conform to service providers, geo-spatial broker conforms to a service broker, and web-based clients conform to service requestors. Fig. 3 shows overall configuration of the suggested framework running on Web Service environment where right bottom side describes the detail of the servers, right top side describes the detail of the broker, and left top side describes clients.

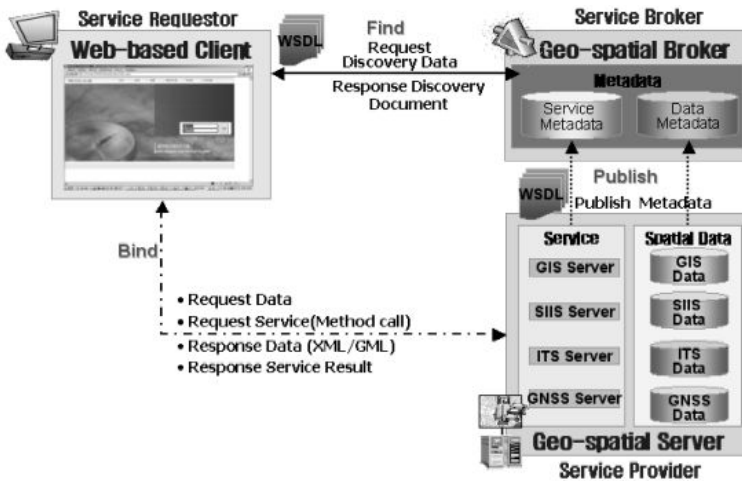


Fig. 3. Overall Configuration of Web Service Framework for Geo-spatial Services.

4.1 Geo-spatial Servers

The framework includes four kinds of servers for the purpose of providing various kinds of geo-spatial services. As shown in Fig. 3, the framework defines GIS, SIIS, ITS, and GNSS servers that are capable of providing vector-typed map, satellite imagery information, real-time traffic information, and location information of moving object, respectively. Also, the framework defines a broker that is able to manage all metadata published by servers. It is remarkable for a broker to be capable of managing geo-spatial functions as well as geo-spatial data. Therefore, servers can publish metadata for all dataset that they can provide, and can publish metadata for all functions that they can serve. In the framework, GIS server publishes GML-typed map and WFS interfaces as example geo-spatial functions. SIIS server publishes GeoTIFF-typed map and WCS interfaces. ITS server and GNSS server only publish functions that can provide real-time traffic information and real-time location information, respectively. In short, our framework defines that GIS server and SIIS server publish both data and functions, ITS server and GNSS server only publish functions. Defining servers and a broker, it is remarkable to accommodate OGC specifications in order to guarantee the conformance to international standards.

As explained in section 3, major problems in designing the Web Service framework are huge volume of geo-spatial data and time-consuming GML conversion, especially in GIS server. Because the problems cause GIS server to consume too much time, it is almost impossible to satisfy clients' request in given time. So, we propose MM server that can satisfy the time limits of clients to some degree. The MM server has two main features: main memory-based geo-spatial data management and preprocessing-based GML conversion.

The MM server controls huge volume of physical main memory about more than 4 GBytes. The MM server converts all geo-spatial dataset into GML documents as soon as it loads dataset from GIS data source. Then, the GML documents always reside in main memory and are directly provided to the clients upon request. Moreover, the framework defines a geo-spatial engine for the MM server to enhance the performance of the server. Exploitation of the MM server can highly enhance the overall performance of the framework by removing problems of huge volume of data and time-consuming conversion. As a result, it becomes possible to serve huge volume of geo-spatial data that has constraints in time limits on Web Service environment. The MM server is composed of *spatial data provider* (SDP), *spatial data manager* (SDM), *spatial data configuration tool* (SCT), and *spatial query manager* (SQM), as shown in Fig. 4.

Spatial Data Provider (SDP). First, the MM server loads vector-typed dataset to main memory using SDP component. The SDP that is a part of SFS specification makes it possible to access any kinds of GIS databases with the same interfaces and methods. It means that the MM server has a distinctive feature of extensibility that can load any kinds of geo-spatial data if there exist SDP components. In the suggested framework, we provide five SDP components

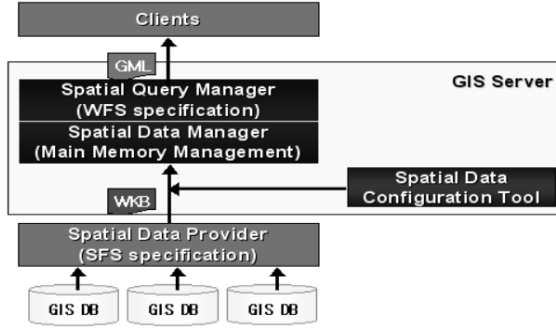


Fig. 4. Architecture of Main Memory-based GIS Server (MM Server).

for different kinds of data sources: shape file, DXF file, SDE spatial engine, ZEUS spatial DBMS, and GeoMania spatial server.

Spatial Data Configuration Tool (SCT). SCT is a simple administration tool for the MM server. This tool invokes the SDM and makes the SDM running state. Then, it chooses SDP component what server wants to load among various kinds of data sources.

Spatial Data Manager (SDM). This is a main module of the MM server. SDM plays an important role in enhancing the performance of the server, for which SDM includes functions such as main memory-based geo-spatial data management and preprocessing based GML conversion. The SDM directly manages main memory over 4 GBytes by taking an access control away from Operating System. The production of GML documents and their residence in main memory are performed by converting WKB (Well-Known Binary) dataset to GML documents when SCT chooses an SDP for geo-spatial dataset. WKB is a standard format for geo-spatial data supported by SDP component. And also, the SDM should have spatial engine for efficient processing of geo-spatial queries. Actually, the spatial engine has functionality of spatial indexing manager using R*-tree [20] indexing method and spatial operator such as “within”, “intersect”, and “contain”. The main advantage of the SDM is that it can give rapid responses to clients’ requests by saving access time of huge volume of geo-spatial data and conversion time from source to GML documents. However, the SDM implemented in this paper has the advantage only for GIS data. Furthermore, the SDM has a disadvantage that it needs much time to make all GML documents reside in main memory when the MM server starts.

Spatial Query Manager (SQM). SQM accepts clients’ requests, analyzes the requests, divides the requests into small queries, and delivers the small queries to SDM. The framework basically accommodates WFS to design the SQM. The processes of analysis of requests, division of requests, delivering small queries

are actually designed by using WFS interfaces such as “*GetCapabilities*”, “*DescribeFeatureType*”, and “*GetFeature*”. The SQM should systematically work with the SDM in order to optimize the overall performance of the MM server. The SQM is physically implemented by using “*Internet Server Application Programming Interface*” (ISAPI) extension component, which guarantees more efficient services than CGI especially when many clients request services simultaneously.

4.2 Geo-spatial Broker

The second major part of the suggested framework is a geo-spatial broker, which conforms to a service broker in Web Service environment between service requestor and service provider. It is the most important purpose of the broker to register and provide metadata for various kinds of geo-spatial services with standardized interfaces. For this purpose, the framework basically adopts WRS. However, the initial WRS does not consider metadata registration and finding using SOAP [21], UDDI, and WSDL that are required in Web Service for geo-spatial functions. Therefore, we should extend the WRS using Web Service architecture. Fig. 5 shows newly suggested architecture of the geo-spatial broker, which is composed of *interface component*, *XML parser component*, *search component*, and *core component*. Interface component as a gate to the broker should provide how clients and servers can access the broker. XML parser component should be able to decode XML documents and encode broker’s results into XML documents. Search component should be able to find out metadata for geo-spatial services and send the results through XML parser component and interface component. Main component of the broker is core component, which directly processes requests of clients and servers. The core component should be composed of WSDL manager, authentication manager, requestor manager, service manager, and data manager for achieving Web Service for geo-spatial data and functions. Authentication manager, requestor manager, and data manager that process metadata for geo-spatial dataset are conceptually defined in

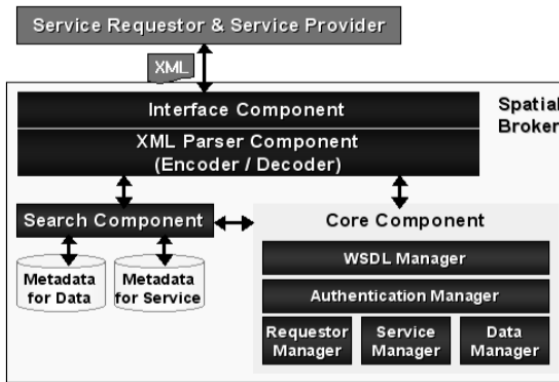


Fig. 5. Architecture of Geo-spatial Broker.

WRS. However, because WSDL manager and services manager are not defined in WRS, the broker defines them to process geo-spatial functions. Authentication manager only checks whether clients or servers are certified users. Requestor manager maintains information for users currently connected to the broker. Data and service manager performs the “Publish” processing for geo-spatial data and functions, respectively. WSDL manager analyzes clients’ requests given as WSDL type and sends the results to requestor manager, service manager, or data manager depending on the requests. The main advantage of the geo-spatial broker is that the broker can handle metadata for both geo-spatial data and geo-spatial functions by extending the WRS.

4.3 Experimental Results of the Main Memory-Based GIS Server

We now present simple performance study to verify the feasibility of the MM server suggested in 4.1. Before presenting the experimental result, we inform you that performance comparison between MM server and ordinary GIS server is omitted. We think it is meaningless to do such experiment, because main memory is always 10 to 20 times faster than secondary memory. So, we only focus on the experiment about the service availability for huge GML documents on wired and wireless network environment. The experiment was performed using PDA on wireless network environment and laptop computer on wired network environment.

We performed sample queries that request variable size of GML documents on PDA and laptop computer. For more exact analysis, we measured loading time, parsing time, and drawing time of GML. The loading time means download time of GML documents from server to client, the parsing time means analysis time of GML documents, and the drawing time means mapping time in client. The experiment was performed on a PDA whose specifications are Intel PXA 250 applications processor (400MHz) chip, 64MB SDRAM, 802.11b Wi-Fi wireless network card of 11Mbps on Microsoft Pocket PC 2002 operating system, and on laptop computer whose specifications are Intel Centrino Mobile 1.6GHz CPU, 1GB DDR SDRAM, 802.11b Wi-Fi wireless network card of 11Mbps. We used real dataset of Seoul district in Korea for the experiment, and we will show its example view of the dataset in section 5. Fig. 6 shows the comparisons about (a) loading time, (b) parsing time, and (c) drawing time. According to the experiment, especially, the loading time and the parsing time on PDA increase exponentially as the size of GML document increases. For the largest GML documents in this experiment (1,629 Kbytes), it took even about 21 seconds to load GML document on PDA, while it took about 3 seconds on laptop computer. It means that the MM server took less than 3 seconds to build GML documents of 1,629 Kbytes that hold about 3,000 objects of line string type. Also, in the parsing time, it took about 33 seconds to parse the GML document of 1,629 Kbytes on PDA and about 1 second to parse the same document on laptop (the difference of parsing time is due to computing power of client, not the performance of MM server). The drawing time shows relatively small difference in the performance (3 seconds on PDA and 0.8 second on laptop). According to the

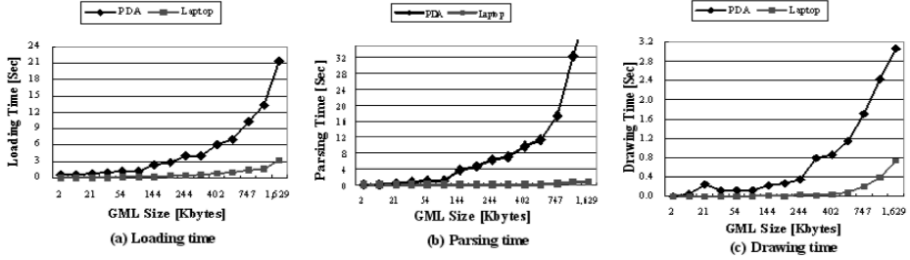


Fig. 6. Performance Comparison on PDA and Laptop.

experiment, it is meaningful to use the MM server when we provide huge volume of GML documents on wired network. However, in case of PDA, it took over 53 seconds in loading and parsing GML document of 1,629 Kbytes (Fig. 6), so we can assert that it is very difficult to provide huge volume of GML documents on wireless network, even though we use the MM server.

As a conclusion, we remark that: (1) the MM server is very feasible to provide huge volume of GML on wired network environment, and (2) the PDA may have problems to receive huge volume of GML on account of wireless network bandwidth and weak computing power, even though the MM server enhance the performance.

5 Implementation

We implemented a prototype for the suggested Web Service framework. To evaluate the framework on Web Service architecture, first of all, we implemented the MM server and then made simple SIIS server, ITS server, and GNSS server as a prototype. The MM server provides vector map of Seoul district in Korea, the SIIS server provides IKONOS information of the same area, the ITS server provides real-time velocity with the type of (Link ID, Velocity) of the same area, and the GNSS server provides location of moving objects with type of (Object ID, Location) of the same area. WFS and WCS are also implemented as a part of the MM server and the SIIS server, respectively. And we implemented the geo-spatial broker and sample client. The geo-spatial broker was implemented by accommodating WRS and by referencing UDDI, and the sample client was implemented to be able to visualize all results format such as JPG, GIF, GML, and GeoTIFF that the servers give at the same time on Web browser. Further, we implemented an administration tool for the geo-spatial broker with the sample client system using Web browser. All parts of the framework are implemented using various programming tools such as Visual C, C#, COM, and so on. Fig. 7 and 8 show our implementation examples for testing the suggested framework.



Fig. 7. Web Service for geo-spatial functions (WFS functions)
“http://www.4s.re.kr/WRS/webapplication/main/main_search_data.html”.



Fig. 8. Web Service for geo-spatial data (GML and GeoTIFF data)
“http://www.4s.re.kr/WRS/webapplication/main/main_search_data.html”.

6 Conclusion

It is difficult to serve complex and huge volume of geo-spatial information on Web Service, because general geo-spatial servers that provide various kinds of geo-spatial data such as GML and GeoTIFF consume too much time in loading and converting huge volume of data to Web Service format. So, we proposed the Web Service framework that has the following features: (1) it can efficiently provide the integrated services for four kinds of geo-spatial servers in standardized manner; (2) it can provide geo-spatial functions as well as geo-spatial data; and (3) it can provide huge volume of GML documents within the allowed time

given by client. And the framework is composed of geo-spatial servers, geo-spatial broker, and web-based client. Speaking in detail, we (1) designed the basic framework that accommodates the OGC specifications in order to provide the integrated geo-spatial data in the standardized manner; (2) extended the basic framework by referencing the W3C specifications in order to serve geo-spatial functions as well as geo-spatial data; and (3) proposed the MM server that can rapidly serve huge volume of GML documents in order to solve the performance problem. And we presented performance comparison between PDA on wireless network and laptop on wired network in the experiment for evaluating the performance of the MM server. The experimental results showed that the MM server is very feasible to provide huge volume of GML on wired network environment, but not on wireless network environment.

Summarizing our work, we designed and implemented the Web Service framework for various geo-spatial services, and we implemented the MM server that can handle huge volume of GML document. We think this framework can be practically applied to Web Service application field for geo-spatial services as a standard model. For example, distributed geolibraries [22] that should provide various kinds of geo-spatial services is able to apply this framework.

Although we did not design and implement the SIIS server, ITS server, and GNSS server that can serve huge volume of data, we can enhance their performance by applying the MM server proposed for GIS server. As the future work, we are planning to update the MM server so that it can work on wireless network environment, for which we consider loading WKB in main memory instead of GML.

References

1. Michael Worboys and Matt Duckham, Integrating Spatio-thematic Information, Proc. GIScience'2002, LNCS 2478, PP.346-361
2. Ming-Hsiang Tsou, An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet, Proc. GIScience'2002, LNCS 2478, PP.313-332
3. Barbara P. Buttenfield, Transmitting Vector Geospatial Data across the Internet, Proc. GIScience'2002, LNCS 2478, PP.51-64
4. Guoray Cai, GeoVSM: An Integrated Retrieval Model for Geographic Information, Proc. GIScience'2002, LNCS 2478, PP.65-79
5. Buttenfield, B, Sharing Vector Geospatial Data on the Internet, Conf. the international Cartographic Association, 1999, PP.35-44
6. Bertolotto, M. and Egenhofer, M., Progressive Transmission of Vector Map Data over the World Wide Web, Proc. GeoInformatica'2001, vol. 5, PP.345-373
7. Laurini, R. and D. Thompson, Fundamentals of Spatial Information Systems, London, Academic Press, 1992
8. Artur Rocha, Joao Correia Lopes, Luis Bartolo and Marco Amaro Oliveria, An Interoperable GIS Solution for the Public Administration, Proc. EGOV'2003, LNCS 2739, PP.345-350
9. OpenGIS Consortium Inc, Simple Features Specification For OLE/COM Specification, version 1.1, 18-May 1999

10. OpenGIS Consortium Inc, Web Map Service Implementation Specification, version 1.1.1, 16-January 2002
11. OpenGIS Consortium Inc, Web Feature Service Implementation Specification, version 1.0.0, 19-September 2002
12. OpenGIS Consortium Inc, Web Coverage Service Implementation Specification, version 0.9, 18-December 2002
13. OpenGIS Consortium Inc, Geography Markup Language Implementation Specification, version 2.1.2, 17-September 2002
14. OpenGIS Consortium Inc, Web Registry Service Implementation Specification, version 0.7, 18-January 2003
15. W3C Consortium Web Services Description Language version 1.2, 11-June 2003
16. OpenGIS Consortium Inc, Catalog Services Specification, version 1.1, 28-March 2001
17. OpenGIS Consortium Inc, OpenGIS Reference Model, version 0.1.2, 04-March 2003
18. OpenGIS Consortium Inc, Styled Layer Descriptor Implementation Specification, version 1.0.0, 19-September 2002
19. Barcay T., Gray J., and Slutz D., Microsoft TerraServer: A Spatial Data Warehouse, Proc. SIGMOD'2000, PP.307-318
20. N.B., H.-P. Kiregel, R.Schneider, and B. Seeger, The R*-tree: An Efficient and Robust Access method for Points and Rectangles", Proc. SIGMOD'1991, PP.322-331
21. W3C Consortium, Simple Object Access Protocol, version 1.2, 24-June 2003
22. National-Research-Council, Distributed Geolibraries-Spatial Information Resources, Washington DC, Mapping Science Committee, 1999
23. Ana Maria de C. Moura, Marcio Victorino, and Aterio Tanaka, Combining Mediator and Data Warehouse Technologies for Developing Environmental Decision Support Systems, Proc. GIScience'2002, LNCS 2478, PP.196-208
24. Larson, R., Geographic Information Retrieval and Spatial Browsing, Proc. GIS and Libraries'1995, PP.196-208

Temporal and Spatial Attribute Extraction from Web Documents and Time-Specific Regional Web Search System

Taro Tezuka and Katsumi Tanaka

Kyoto University
Graduate School of Informatics
{tezuka, tanaka}@dl.kuis.kyoto-u.ac.jp

Abstract. Regional web search engines provide users with spatially restricted web contents. We propose a time-specific regional web search engine, where users can retrieve both spatially and temporally restricted information from the Web, i.e. a description of a shop available at certain time in a certain region. Because there are many web pages containing descriptions of more than one geographic entity, a matching mechanism of spatial and temporal attributes is required. We used HTML tag structures to make matches between addresses and temporal interval expressions. Based on the extracted result, map-based visual interface was implemented.

1 Introduction

The Web is now widely used for retrieving **regional** information, in spite of the fact that it is a **global** information system. Various regional web search systems have been proposed that enable users to retrieve information related to a specific geographic region. We propose a **time-specific regional web search system**, where the user can specify both the temporal and spatial attributes in addition to spatial attributes of the geographic entities.

The system can be used when the user wants to know if a shop is open at a specific time in a specific region. Some of the motivating examples are listed below:

- 8:00** Find a hamburger stand open early in the morning, near the highway.
- 12:00** Find a restaurant, open during lunch break, near the office.
- 19:00** Find a gift shop, open in the evening, by the station.
- 22:00** Find a nightclub, downtown.
- 1:00** Find a bar, open after midnight.

Many regional search systems, (not necessarily web-based), do not accept temporal restrictions in a query. One reason is because the map data provided by GIS suppliers does not contain business hour information. Finding business hours for each shop is a labor-intensive task, and it is unlikely that GIS suppliers will provide such data in the near future.

Currently, there is a growing amount of business hour information available on the web sites. However, with since information are scattered over the Web, the user must check a considerable number of web pages to obtain the desired information.

For example, the user must take the following steps to reach to a description of a shop, open at a specific time in a specific region.

1. Think of a region name designating the target area.
2. Send a query 'region name + shop category name' to a search engine.
3. Retrieve the results and check each page
4. See if temporal interval expressions are included, and, if so, check to see if they fulfill the requirements.

If the spatial and temporal attributes were collected prior to the search, and presented on a map-based visual interface, the user's load would be greatly reduced.

The main objective of our research is to construct a system for the automatic extraction of the spatial and temporal attributes from the web contents and to be able to visualize the map-based interface.

The basic architecture of the system is shown in Figure 1.

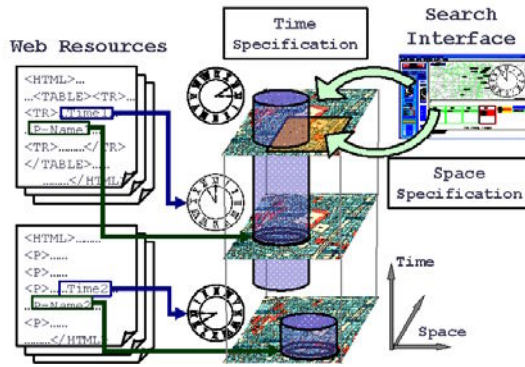


Fig. 1. The Extraction and Visualization of Temporal and Spatial Attributes.

In Section 2, we will discuss some related research. In Section 3, we define some of the terms used in this research. Then, in Section 4, we will describe algorithms used for the extraction and the matching of the attributes. In Section 5, we will evaluate the algorithms. In Section 6 we will describe an implementation an enhanced dynamic map interface for the spatial and temporal specifications of web contents, and, in Section 7, we will conclude our presentation.

2 Related Work

Sagara et al. have developed GeoParse [4], that extracts addresses from web content and converts them to coordinates. Yokoji et al. implemented “Kokono Search” [3], where users can search web content related to specified coordinates, addresses, or station names. These systems retrieve spatially specific web contents, whereas in our system, users can search for both spatially and temporally specific information on the Web.

Sumiya and Tanaka have discussed the management of web contents as a web archive system [9]. This research focused on the temporal attributes of the web pages themselves, and not on the geographic entities described within those pages.

Spatially and temporally specific retrieval of web content are also discussed, in terms of the standardization of temporal and spatial attributes in XML format [6]. Point Of Interest eXchange Language Specification (POIX) [7] is a proposed specification for the location and temporal attributes in web documents. Based on this standardization, some works describe the extraction of spatial and temporal attributes [8]. However, at this moment, most web pages are not based on these specifications. Addresses and business hours are usually expressed in plain text. Therefore, the text-mining approach is still useful for retrieving information from most web contents. Also, because so many different standards have been proposed, it may take a long time to agree on a universal standard.

There are also web sites that provide time-specific searches for shops in this type of database must be manually constructed. In our system, time attributes are automatically collected from the Web.

Kiyomitsu et al. discussed the personalization of web content, based on the date and the location of the user's access [10]. Although they used temporal and spatial attributes in their analysis, their goal was solely to personalize the web contents. In their system, the users cannot arbitrarily specify the time and region.

Morishita et al. developed SpaceTag [11], where the user's mobile device can receive different information, depending on his/her location. The information had to be manually registered, either by the user or an administrator. Automatic extraction from the Web was not considered. In our research, no manual registration is necessary, and even shops that were not included in the original map data can be added to the database, once their spatial and temporal attributes are given on a web site.

Hirata and Murakami discussed the Memory-Organizer [12], which is a system for helping users' personal memories by linking memorandum to certain locations and times. They briefly discussed the extraction of event schedules and shop business hours from the Web. However, Memory-Organizer is not a search system and it does not have a visualized interface that allows users to specify the region and time. Additionally, they did not implement the actual web mining system for Memory-Organizer.

Our method of extracting matches between the spatial and temporal attributes, using HTML tags, resembles that of Hattori et al. [14]. However, their system was for matching image files along with their descriptions.

3 Definitions

The terms used in this research are defined as follows.

Geographic entities are physically continuous entities with static positions in geographic space, in/on which people can take action. Shops are geographic entities, as are stations, buildings, sight-seeing spots, and landmarks. Because our research deals with the available hours associated with geographic entities, locations related to social activities will be our primary focus.

An **address** is an administrative district, in which geographic entities are located. The required level of specification for addresses differs by application, and, in this research we used minor-sub-districts, which are sub-division of wards. The spatial attributes extracted from the Web are limited to addresses in this research. The number of addresses contained on a web page p is expressed by $a(p)$.

A **temporal interval**, is a stretch of time, related to a geographic entity. In most cases, this is a stretch of time during which the geographic entity is available. For shops, it is their business hours. For public facilities, it is hours they are available. For events, it is time interval the event is held. The temporal attributes extracted from the Web are limited to temporal intervals, in this research. They have the advantage of reducing noise, since collecting time expression results in a lot of noise, due to the ambiguity in its expressions. The number of temporal intervals contained in a web page p will be expressed as $t(p)$.

A **1-1 match** is a state where one address and one temporal interval are found in a specific location on a web page. The methods used for obtaining 1-1 matches will be discussed in detail later.

4 Extraction Algorithm

The extraction algorithm consists of three parts: temporal interval extraction, address extraction, and a matching of the two.

4.1 Temporal Interval Extraction

Temporal intervals of geographic entities are expressed in various forms, including these shown in the list below.

- From 9:00 a.m. to 5:00 p.m.
- 10:00-18:00
- From 5:00 to midnight
- P.M. 18:00 - A.M. 2:00
- 21:00-2:00

Yet there are certain patterns in these forms that we can automatically convert to a pair of opening and closing time. Below is an example of these conversion patterns.

- (PM|P.M.) $h : m \longrightarrow h = h + 12$
- $h : m$ (PM|P.M) $\longrightarrow h = h + 12$
- $h > 24 \longrightarrow h = h - 24$
- Noon $\longrightarrow h = 12, m = 0$
- $h_1 : m_1$ (to|-) $h_2 : m_2 \longrightarrow \text{OpenAt}=h_1 : m_1, \text{CloseAt}=h_2 : m_2$

We have constructed a regular expression that represents the above patterns.

There are cases where a shop's business hours depend on the day of the week. Also, some shops specify lunch hours in addition to business hours. Other pages write the morning and afternoon hours separately. If we consider all these factors, the possible permutation will be enormous. In our research, we will use a pair that forms a 1-1 match only, and avoid these complications.

If time expressions were extracted instead of temporal interval expressions, there would be a large amount of noise, due to the ambiguities in the format for the time expressions. For example, if we were to extract “3 to 4 digits separated by a colon”, or as a regular expression ($/[0-2]?[0-9] : [0-5][0-9]/$), we would have difficulty selecting time expressions from the unexpected matches. Because we have limited our retrieval to temporal intervals, the noise problem is greatly reduced.

4.2 Address Extraction

All that is required to extract addresses from web contents is to match the required address with the addresses stored in the database. Unlike the extraction of temporal intervals, no special regular expressions are necessary. However, a few cases require special care.

- Ambiguities in address expressions. There are cases where a city/ward name and a district name are separated by a street name or a route direction.
- The city/ward name and the district name are written separately in the same document. Although this is not the formal way of expressing an address, there is a chance that they will correspond, and the location can be specified.
- A building or an organization name followed by a ward name enclosed in parentheses. This expression is often found in local newspaper articles. If this is used together with building name database, the location can be specified.

Using address and coordinate data from GIS, the address is converted into coordinates and stored in the database.

4.3 Matching Algorithm

Let P indicate a set of web pages, and p an arbitrary web page in P . p can contain descriptions of an arbitrary number of geographic entities.

If p contains only one address and one temporal interval, then it is likely that those two attributes are associated with an identical geographic entity. We call this a **page-wise 1-1 match** between the address and the temporal interval.

On the other hand, there are cases where a web page contains more than one address or temporal interval, i.e. a list of shops' information. To extract information from such web pages, a different method to retrieve a 1-1 match must be considered.

We use HTML tag structures to retrieve 1-1 matches from this type of web pages. HTML structures, unlike XMLs with stricter machine-readable formats, allow start tags without end tags. Thus the tree structure is harder to extract from HTML documents compared to XML documents. Therefore, we extract in the following way, instead of extracting tree structure.

If any string wedged between $\langle T \rangle$ and $\langle T \rangle$, or $\langle T \rangle$ and $\langle /T \rangle$, contains one address and one temporal interval, we call this a **tag-wise 1-1 match**. A pseudo-code for our extraction mechanism is as follows.

PARSE the target web page.

PUSH each start tag, address, and temporal interval to the stack *PED*.

STORE the most recent address into *RA*.

STORE the most recent temporal interval into *RT*.

IF the parser reaches a start/end tag of type *T*,

SCAN through each entry *AN* of *PED* from top to bottom, ignoring the very top entry.

IF *AN* is an address, *CA* ++.

IF *AN* is a temporal interval, *CT* ++.

IF *AN* is a type *T* start tag,

IF $CA = 1 \wedge CT = 1$,

OUTPUT the pair (*RA*, *RT*) as a tag-wise 1-1 match.

ERASE all addresses and temporal intervals in *PED*.

Figure 2 shows the flow-chart for retrieving tag-wise 1-1 matches.

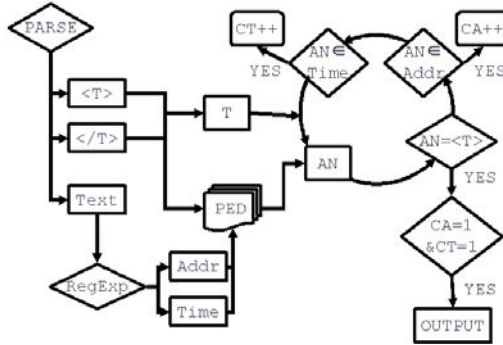


Fig. 2. Extraction of Tag-wise 1-1 Matches.

This algorithm ensures that the system will capture 1-1 matches for addresses and temporal intervals, wedged between $\langle T \rangle$ and $\langle /T \rangle$, and between also $\langle T \rangle$ and $\langle T \rangle$. The latter case is where the end tag for the start tag $\langle T \rangle$ is missing, yet the start tag of the same type follows, such as $\langle TR \rangle$ followed by another $\langle TR \rangle$, or $\langle P \rangle$ followed by another $\langle P \rangle$. These formats are often used in the list of geographic entities. Once the system captures a tag-wise 1-1 match, all addresses and temporal intervals in the *PED* stack are erased, avoiding matches between expressions that are far apart.

Lists of shop data are often organized using $\langle TABLE \rangle$ tags, with each entry divided by $\langle TR \rangle$ tags. There are also cases where each entry is separated by $\langle P \rangle$ tags. In both cases, if each entry contains one address and one temporal interval, a tag-wise 1-1 match can be extracted.

In the following section, the results of the page-wise and tag-wise 1-1 match extractions are compared.

5 Evaluation

5.1 Original Data

The data used for the evaluation consists of a set of place names and web pages. The set of place names was taken from a digitalized residential map of Kyoto, Japan, provided by Zenrin Ltd. [15]. Web pages were collected using a web crawler. Because our intention was to build a regional web search system, we used a focused web crawler. A focused web crawler, unlike regular web crawlers, does not store all the web pages it can access. Instead, it stores only those pages that fulfill certain conditions specified by the user, and then it traces links from those pages. Focused web crawlers are more efficient for collecting web pages related to a specific topic. This method is based on the assumption that a page related to a specific topic is more likely to be linked to another page related to the same topic. By using a focused web crawler, the system no longer needs to search the entire Web and store vast number of web pages, just to gather information on a specific topic. A lot of research indicates that focused crawlers are more efficient for collecting web pages related to specific topics [3][16][17]. In our experiment, the condition used for filtering the web pages was that the page must contain an arbitrary place name in the target area (Kyoto). We have collected 157,297 web pages and used them for the evaluation of our temporal information extraction.

5.2 Recall and Precision

Figure 3 shows compositions of addresses and temporal intervals from the collected web pages. We checked 100,000 pages, by increments of 1,000. Each line indicates following types of compositions: **1.** contains more than one address and temporal interval. ($a(p) > 0 \wedge t(p) > 0$), **2.** contains one address and more than one temporal interval. ($a(p) = 1 \wedge t(p) > 0$), **3.** contains one temporal interval and more than one addresses. ($a(p) > 0 \wedge t(p) = 1$), **4.** contains one address and one temporal interval. ($a(p) = 1 \wedge t(p) = 1$). Because we used a focused web crawler to collect web pages, they were more likely to contain addresses, compared to randomly collected web pages.

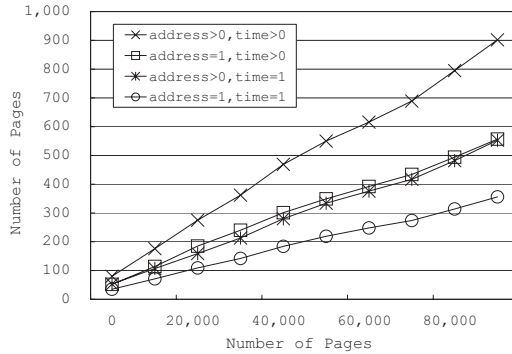


Fig. 3. The Numbers of Addresses and Temporal Intervals in a Web Page.

Figure 3 shows that a considerable number of web pages contain more than one address or temporal interval; therefore, a page-wise 1-1 match can retrieve only a small number of possible matches.

The bar graph in Figure 4 shows the recall of matching pairs from 157,297 web pages, using page-wise and tag-wise 1-1 matches. The bar for the tag-wise 1-1 match also shows the type of tag used for matching.

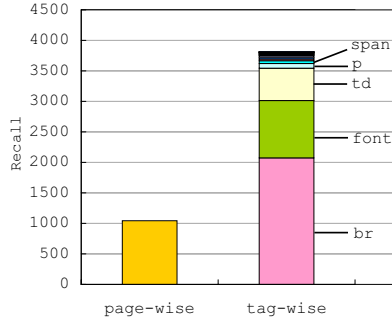


Fig. 4. Recall for Page-wise and Tag-wise 1-1 Matches.

We obtained 1,044 pairs by page-wise 1-1 matching. In tag-wise 1-1 matching, we obtained 3,814 pairs. Tag-wise 1-1 matching gives about 3.7 times more 1-1 pairs, compared to page-wise 1-1 matching.

We also measured the precision. This is the ratio of correct 1-1 matches from the retrieved 1-1 matches, where the correct 1-1 match is when the address and the temporal interval designate an identical geographic entity. This check was performed manually. The result is shown in Figure 5.

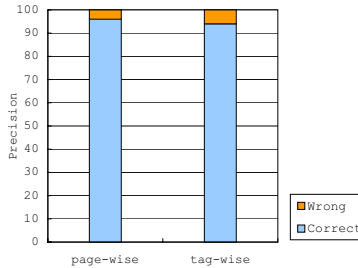


Fig. 5. Precision for Page-wise and Tag-wise 1-1 Matches.

The precision for the page-wise 1-1 match was 96%, while the precision for the tag-wise 1-1 match was 94%. It indicates that large proportion of address and temporal interval pairs designate identical geographic entities. Pairs that designated different entities resulted from the following reasons:

Event: The event time + the organizer’s address

Notice: Relevant time + the contact address

Public Transportation: Available time + the office address

However, proportion of such incorrect 1-1 matches are small.

Figure 6 shows the difference in processing times between the extraction of page-wise and tag-wise 1-1 matches. The experiment was performed on 100,000 web pages, by increment of 10,000 pages. The result shows that the difference in processing time is not significant.

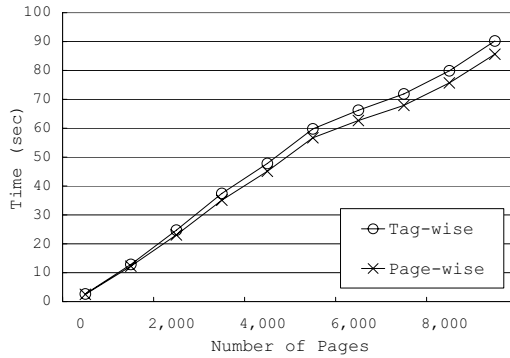


Fig. 6. Comparison of Processing Times.

6 Implementation

Based on the algorithm described in the previous sections, we implemented a time-specific regional search system **ChronoSearch** (Figure 7). ChronoSearch has a client-server architecture, using Java Applet, Tomcat Servlet, and Postgres Database Management System.

The user can specify an arbitrary region, using the map interface, and specify time by the clock interface, which is based on a real-world metaphor. Titles of the retrieved web contents are shown on the map interface. These are also hyperlinks to the content itself.

If the user specifies a word in the keyword form, the result is filtered to those containing the keyword within their title. The following list shows how geographic entities are filtered by their attributes, using the ChronoSearch components.

Web Content	Spatial	→ Address	↔ Map Interface
	Temporal	→ Temporal Interval	↔ Clock Interface
	Categorical	→ Title	↔ Keyword Form

The system not only provides the user with a means of searching for information, but also a visualization of how the city activities shifts as the time passes.

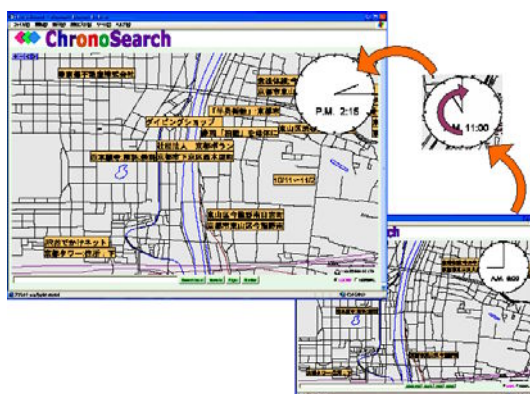


Fig. 7. ChronoSearch: a Time-Specific Regional Search System.

Unlike our map-based visualization, traditional paper-based guidebooks will have a lot of difficulty presenting spatially and temporally restricted geographic entities. Preparing different maps for each major point in time is impractical. Our system uses the capability of digitalized map rendering to realize this function.

7 Conclusion

In this research, we developed a method for extracting the spatial and temporal attributes of geographic entities from the Web. The evaluation compared tag-wise 1-1 matches with page-wise 1-1 matches.

Our future research will include following topics.

Because the extraction of addresses and temporal intervals involves ambiguities, we may consider using factors to express quality of the matching. This can be visualized on the map interface by colors. This will reduce confusion caused by incorrect 1-1 matches.

In the present system, we are using addresses only to extract the spatial attributes. In the future, we may use building and shop names as well, based on residential map data provided by GIS suppliers. This may result in the retrieval of more related web content. In this research, we have used temporal intervals on a scale within one day; in the future, we want to include longer temporal intervals, i.e. a day of the week, a month, a season, or a year. Seasons might be of special interest to tourists. Content filtered by days of the week can be used to show shifts in thriving districts.

Also in the present system, the user can specify a point in time only. The temporal logic, described by Allen [18], may help extend our system by allowing temporal interval specifications.

Although we have implemented our system using a set of web pages collected by a web crawler, we could also use a meta search engine. In that case the addresses and landmark names, shown on the map interface, will be sent to the search engine, and the results will be parsed, filtered by the temporal interval and the keywords specified by the user.

Acknowledgment

This paper is supported in part by The Special Research Area's Grant In Aid For Scientific Research(2)for the year 2004 under the project title "Research for New Search Service Methods Based on the Web's Semantic Structure" (Project No: 16016247, Representative: Katsumi Tanaka). This paper is also supported in part by Informatics Research Center for the Development of Knowledge Society Infrastructures (COE program of the Ministry of Education, Culture, Sports, Science and Technology, Japan).

References

1. K. Hiramatsu, K. Kobayashi, B. Benjamin, T. Ishida, and J. Akahani, "Map-based User Interface for Digital City Kyoto", in Proceedings of the 10th Annual Internet Society Conference, Yokohama, Japan, 2000
2. E.P. Lim, D. Goh, Z. Liu, W.K. Ng, C. Khoo, S.E. Higgins, "G-Portal: A Map-based Digital Library for Distributed Geospatial and Georeferenced Resources," in proceedings of the 2nd ACM+IEEE Joint Conference on Digital Libraries (JCDL 2002), Portland, Oregon, 2002
3. S. Yokoji, K. Takahashi, N. Miura, K. Shima, "Location Oriented Collection, Structuring, and Searching Methods of Information," Information Processing Society of Japan Journal, Vol. 41, No. 7, pp. 1987-1998, 2000
4. T. Sagara, M. Arikawa, M. Sakauchi, "Spatial Information Extraction System Using Geo-Reference Information," Information Processing Society of Japan Journal:Database, Vol.41, No.SIG6(TOD7), pp.69-80, 2000
5. A. G. Woodruff and C. Plaunt, "GIPSY: Automated geographic indexing of text documents," Journal of the American Society for Information Science, Vol. 45, No. 9, pp.645-655, 1994
6. A. Schmidt and C. S. Jensen, "Spatio Temporal Data Exchange Standards," IEEE Data Engineering Bulletin, Vol. 26, No. 2, pp. 50-54, 2003
7. H. Kanemitsu and T. Kamada, "POIX: Point Of Interest eXchange Language Specification," available at <http://www.w3.org/TR/poix/>, 1999
8. M. Matsudaira, T. Ueda, M. Fuchigami, H. Oonuma, Y. Morita, "Extraction of Keywords from Documents and Collection of Related Information," The Japanese Society for Artificial Intelligence Special Interest Group on Semantic Web and Ontology (SIG-SWO), 5th Meeting, Document No. 2, pp.1-6, 2004
9. K. Sumiya and K. Tanaka, "Time Information Management for Web Archives and its Application," Information Processing Society of Japan SIGNotes, Vol.2003, No.71, 2003-DBS-131(II)-85, pp.109-116, 2003
10. H. Kiyomitsu, A. Takeuchi, K. Tanaka, "Web Reconfiguration by Spatio-Temporal Page Personalization Rules Based on Access Histories," Proc. of the Symposium on Applications and the Internet (SAINT 2001), pp.75-82, IEEE Press, 2001
11. K. Morishita, M. Nakao, H. Tarumi, Y. Kambayashi, "Time Specific Object System: Design and Implementation of SpaceTag Prototype System", Information Processing Society of Japan Journal, Vol. 41, No. 10, pp. 2689-2697, 2000
12. T. Hirata and H. Murakami, "A Prototype for Mobile Memory Construction System," Osaka City University Academic Information Center Bulletin, Vol. 4, pp. 45-50, 2003
13. A. Arasu, H. Garcia-Molina, "Extracting Structured Data from Web Pages,"
14. T. Hattori, I. Sawanaka, A. Nadamoto, K. Tanaka, "Determination of Synchronizable Region for Passive View of the Web and TV-Programming Markup language S-XML," Information Processing Society of Japan SIGNote, Vol.2000, No.44 00-DBS-121-2, pp. 9-16, 2000

15. ZENRIN CO.,LTD, <http://www.zenrin.co.jp/>
16. S. Chakrabarti, M. van den Berg, B. Domc, "Focused crawling: a new approach to topic-specific Web resource discovery," in Proceedings of the 8th International World-Wide Web Conference (WWW8), Toronto, Canada, 1999
17. M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles and M. Gori, "Focused Crawling Using Context Graphs," in Proceedings of the 26th International Conference on Very Large Databases, VLDB 2000, Cairo, Egypt, 2000
18. Allen, J.F. "Maintaining Knowledge about Temporal Intervals," Communications of the ACM, Vol. 26, No. 11, pp. 832-843, 1983
19. T. Tezuka, R. Lee, H. Takakura and Y. Kambayashi, "Cognitive Characterization of Geographic Objects Based on Spatial Descriptions in Web Resources," in Proceedings of the Workshop on Spatial Data and Geographic Information Systems (SpaDaGIS), Milano, Italy, 2003

Broadcasting and Prefetching Schemes for Location Dependent Information Services

KwangJin Park, MoonBae Song, and Chong-Sun Hwang

Dept. of Computer Science and Engineering, Korea University
5-1, Anam-dong, Seongbuk-Ku, Seoul 136-701, Korea
{kjpark,mbsong,hwang}@disys.korea.ac.kr

Abstract. The results of location-dependent queries(LDQ) generally depend on the current locations of query issuers. Many mechanisms, such as broadcast scheme, prefetching scheme, or caching scheme have been developed to improve system performance and provide better service for location dependent information services(LDISs). However, the client's mobility may lead to inconsistency problems. In this paper, we introduce the broadcast-based LDIS scheme(BBS) in the mobile computing environment. In the BBS, broadcasting data items are sorted sequentially based on their location and the server broadcasts the location dependent data(LDD) without additional indices. Then we present a data prefetching scheme and OBC(Object Boundary Circle) in order to reduce the client's tuning time. The performance for the proposed scheme is investigated by various environmental variables such as distributions of the data items, average speeds of the clients and the size of the service area.

1 Introduction

In today's increasingly mobile computing world, people wish to be able to access various kinds of services at any time and in any place. However, the mobile computing environment is characterized by narrow network bandwidth and limited battery power. Furthermore, the changes in locations of the mobile clients can be difficult to handle in an LDIS, particularly in the areas of query processing and cache management[1]. Techniques such as caching, prefetching and broadcasting provide effective means of reducing the wireless bandwidth requirement and can also save the client's battery power consumption. The broadcasting of spatial data together with an index structure is an effective way of disseminating data in a wireless mobile environment [2]. Using an index can help the client to reduce the amount of time spent listening to the broadcast channel. However, the average time elapsed between the request for the data and its receipt may be increased as a result of these additional messages. Therefore, the best access time is obtained when no index is broadcasted along with the file [2]. Thus, reducing the size of the index is an important issue in wireless data broadcasting environments. The value of location dependent data depends on the location.

The answer to a query depends on the geographical location from which the query originates. Let's consider an example in which a salesman drives a car and has to visit all of his customers. The salesman sends a query, such as, "what are the names and addresses of the customers near to my current location?", using his mobile device. Once the salesman gets the answer from the server, he visits the customers in order of increasing distance from his current location. At any one time, the list of clients to visit consists of those customers who have not yet been visited. To handle such a query, the positions of the objects and the clients must be found. A common way to perform a location dependent query processing has two types, namely client initiate approach and server initiate approach. In the client initiate approach, the client sends a location dependent query to the server, and the server sends the result back to the client according to the client location. In contrast to client initiate approach, in the server initiate approach, the server broadcasts location dependent reports to the clients [7]. Since the cost of broadcasting does not depend on the number of users, this method will scale up with no penalty when the number of users grows. Moreover, techniques such as caching and prefetching can save the clients' battery power consumption and reduce the response time. However, contained broadcast messages may not be valid to all the clients. Therefore, how to organize the broadcast report is one of the important issues in server initiate approach. In this paper, we propose broadcast-based LDIS under a geometric location model. We first introduce the broadcast based location dependent data delivery scheme(BBS). In this scheme, the server broadcasts reports, which contains IDs of data items(e.g., building names)and values of location coordinates periodically to the clients. Broadcasting data objects are sorted sequentially based on their location. Then we introduce the prefetching scheme in LDIS for mobile computing environment. To manage the mobile clients' cache, the server does not maintains the information about clients' cached data item. Instead, the server broadcasts data items that are sorted based on the locations of objects, and the client prefetches and caches the data item in anticipation of future accesses. The rest of the paper is organized as follows: Section 2 gives the background of broadcast model and cache maintenance scheme. Section 3 describes the proposed BBS scheme and prefetching method. The performance evaluation is presented in section 4. Finally, section 5 concludes this paper.

2 Background

With the advent of high speed wireless networks and portable devices, data requests based on the location of mobile clients have increased in number. However, there are several challenges to be met in the development of LDISs[3], such as the constraints associated with the mobile environment and the difficulty of taking the user's movement into account. Hence, various techniques have been proposed to overcome these difficulties.

2.1 Broadcast Model

Disseminating data through a broadcast channel allows simultaneous access by an arbitrary number of mobile users and thus allows efficient usage of scarce bandwidth. In [4], they introduce a technique for delivering data objects to the clients in asymmetric environments. In this scheme, groups of pages, such as hot and cold groups, with different broadcast frequencies are multiplexed on the same channel. Then, those items stored on the faster disks are broadcast more often than those items on the slower disks. However, the wireless broadcast environment is affected by the battery power restrictions of the mobile clients. Air indexing is one of techniques that attempts to address this issue, by interleaving indexing information among the broadcast data objects. There are several indexing techniques, such as the distributed indexing approach [2], the signature approach [5] and the hybrid approach [6].

2.2 LDIS Schemes

In the mobile computing environment, caching data on the client's side is a useful technique for improving the performance. However, the frequent disconnection and mobility of the clients may cause cache inconsistency problems. In [7], they propose location dependent cache invalidation schemes for mobile environments. In this scheme, they use bits to indicate whether the data item in the specific area has been changed. Moreover, they organize each service area as a group in order to reduce the overhead for scope information. In [8], they proposed a PE(Polygonal Endpoint) and an AC(Approximate Circle) scheme. The PE scheme records all the endpoints of the polygon representing the valid scope, while the AC scheme uses an inscribed circle from the polygon to represent the valid scope of the data.

3 Proposed Algorithms

In this section, we describe two schemes for LDIS. We first introduce the broadcast-based LDIS scheme(BBS). In this scheme, the server broadcasts reports which contain the IDs of the data objects(e.g., building names)and the values of the location coordinates. The data objects broadcast by the server are sorted based on the locations of the data objects. Then, we present a data prefetching scheme and OBC, in order to reduce the client's tuning time.

3.1 BBS(Broadcast Based LDIS) Scheme

The best access time is obtained when no index is broadcasted along with the file [2]. Let N is the number of objects to be broadcasted. Then, $N/2$ is an average time to get the required data object. Thus, best algorithm for access time is to $N/2$. In BBS method, the sever periodically broadcasts the IDs and the coordinates of the data objects to the clients. The server does not insert an

index in the broadcast cycle. Instead, the broadcast data objects are sorted sequentially according to the locations of the data objects. The proposed scheme gives the fastest access time in LDIS. If there is no index, it requires the client to listen to the broadcast channel N time to handle the NN(nearest neighbor) query. However, the server broadcasting sorted data objects gives only $N/2$ tuning time. Moreover, based on the distance between the data objects, we assign different weight values to each data object by using the OBC(Object Boundary Circle). Then, the data objects can be sent using different broadcast frequencies, by classifying them into hot and cold groups[4]. We discuss this issue in the section concerning the performance evaluation. The following shows comparison of access time and tuning time between BBS and index method[2]. Let m denotes the number of times broadcast indices:

- Access time:
- $\frac{1}{2} * ((m + 1) * index + (\frac{1}{m} + 1) * N)$ (index method)
- $\frac{1}{2} * N$ (BBS method)
- Tuning time:
- $2 + \lceil \log_n(N) \rceil$ (index method)
- $\frac{1}{2} * N$ (BBS method)

As states above, the BBS scheme may have longer tuning time than the index method. Our solution to this problem is partitions of the service area, broadcasting hot data objects and prefetching strategy. In the BBS, the structure of the broadcast affects the distribution of the data objects.

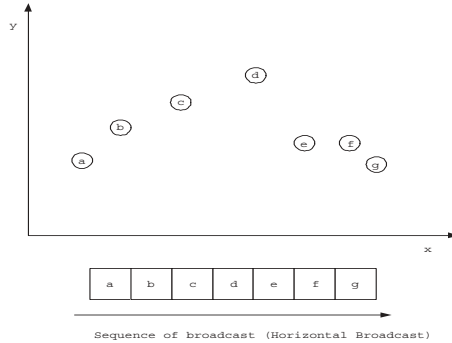


Fig. 1. Example of Horizontal Broadcast.

For example, as shown in Fig 1, if the data objects are horizontally distributed, the server broadcasts data objects sequentially from the leftmost data object to the rightmost data object. A simple sequential broadcast can be generated by linearizing the two dimension coordinates in two different ways: i.e. horizontal broadcast(HB) or vertical broadcast(VB). In HB, the server broadcasts the LDD in horizontal order, that is, from the leftmost coordinate to the

rightmost coordinate. On the other hand, in VB, the server broadcasts the LDD in vertical order, that is, from the bottom coordinate to the top coordinate. In order to decide whether HB or VB, the server uses the following algorithm:

- *Notations*:
- *leftmost_P*: a point that is located at the leftmost extremity in the map(e.g., object 'a' in Fig 1)
- *leftmost_P'*: the x-axis coordinate of a point that is located at the leftmost extremity in the map with the exception of *leftmost_P*, where the value of the coordinates of *leftmost_P' ≥ leftmost_P* (e.g., object 'b' in Fig 1)
- *top_P*: a point that is located at the top of the map(e.g., object 'd' in Fig 1)
- *top_P'*: a point that is located at the top of the map except top_P, where the value of *top_P ≤ top_P'* (e.g., object 'c' in Fig 1)
- *MAX*: maximum number of compares (number_of_objects -1)
- *NOC*: number_of_compares(initial value is 0)
- *x-dist_counter*: sum of distances based on the x_coordinates
- *y-dist_counter*: sum of distances based on the y_coordinates

Algorithm 1. The server decision algorithm for VB or HB data broadcasting

Input: data objects' IDs and locations;

Output: selection result for HB or VB;

```

1:  find leftmost_P
2:  while(NOC <= MAX) {
3:      do : {
4:          find leftmost_P' (if more than two points have same x-axis
                        value, select upper point first)
5:          compare x-dist and y-dist
6:          if x-dist > y-dist
7:              then x-dist_counter ++
8:              else y-dist_counter ++
9:              leftmost_P = leftmost_P'
10:         NOC ++
11:     }
12: }

13: if x-dist_counter > y-dist_counter
14:     then select HB for the broadcast data object
15: else select VB for the broadcast data object

```

In order to identify the nearest object using the BBS the scheme, the client has to compare the most recently delivered object with the previous one during the tuning time. The client uses the following algorithm to identify the nearest object:

- *Notations:*
- S : server data set
- O : broadcast data object, where $O \in S$
- O_c : current broadcast data object, where $O_c \in S$
- O_p : previous broadcast data object, where $O_p \in S$
- O_n : nearest data object
- C_l : client's location

Algorithm 2. The client algorithm used to identify the nearest object

```

1:  for each object  $O \in S$ 
2:  {
3:      do {
4:          compare  $dist(O_c, C_l)$  and  $dist(O_p, C_l)$ 
5:          if ( $dist|O_c - C_l| > dist|O_p - C_l|$ )
6:               $O_n = O_p$ 
7:              return  $O_n$ 
8:          else if ( $(dist|O_c - C_l| < dist|O_p - C_l|)$  and  $O_c$  is
              last data object of the broadcast period)
9:               $O_n = O_c$ 
10:             return  $O_n$ 
11:         else
12:              $O_c = O_p$ 
13:     } while (find out  $O_n$ )
14: }
```

Since it does not have the location information of all of the data objects, the client cannot estimate which data will be broadcast next. Hence, even if the server delivers data objects sequentially based on their coordinate values, it is difficult to determine which data object is the nearest to the client. In our scheme, the client maintains a queue, and determines the size of the window w (hereafter we called w_q) which indicates the number of data objects that will be left in the queue. The client maintains objects in the queue based on the size of w_q and that can be represented as follows:

- *Notations:*
- O_j : an object in the map
- T_o : the timestamp of an object
- T_c : the timestamp of the current broadcasted object
- S : set of objects in the map
- S_q : set of objects in the queue
- w_q : size of the windows in the queue

Then $S_q = \{ \langle O_j, T_o \rangle \mid (O_j \in S) \wedge (T_c - w_q \leq T_o \leq T_c) \}$

3.2 Prefetching Method

In this section, we present a prefetching method for use in LDIS. In this method, the client prefetches the data object for future uses. Let w_p be the size of prefetched data objects. The client adjusts the size of w_p according to the speed and the size of the cache. Let client's current location be point q and the object's location be point p . We denote the Euclidian distance between the two points p and q by $\text{dist}(p, q)$. In the map, we have $\text{dist}(p, q) := \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$

Let $P := \{p_{-n} \dots, p_{-2}, p_{-1}, p_0, p_1, p_2 \dots, p_n\}$ be a set of n distinct points that represent the data objects, and q represents a query point.

– *Notations:*

– $w, n \geq 0$ and $(w - n) \geq 0$

– $\text{target} =$ an object p_0 , where $p_0 \neq p_n$ and $\{p_{-n}, p_0, p_n\} \in P$ then $\text{dist}(p_0, q) \leq \text{dist}(\forall p_{-n}, q)$ or $\text{dist}(p_0, q) \leq \text{dist}(\forall p_n, q)$

– $p_{-max} =$ an object p_{-w} , where $\text{dist}(p_{-(w+n)}, q) \leq \text{dist}(p_{-w}, q)$
 $\leq \text{dist}(p_{-(w-n)}, q)$

– $p_{max} =$ an object p_w , where $\text{dist}(p_{w-n}, q) \leq \text{dist}(p_w, q) \leq \text{dist}(p_{w+n}, q)$

A query can be categorized as the nearest or the k -nearest based on the number of returned objects. The number of returned objects depends on the value of w_p . If we regard the value of w_p as n , the number of returned objects is $2n + 1$. Hence, if the value of n is 3, the number of returned objects is 7 (7-nearest neighbor). In order to adjust the value of k of the k -nearest objects, the proposed scheme simply adjusts the size of w_p . The formal description of the algorithm used for prefetching at the client is as follows:

Algorithm 3. The client algorithm for data prefetching

```

1:  while (a client is looking for the nearest object) {
2:    active mode (listen to the broadcast channel)
3:    if (desired data comes from the server) { // use algorithm 2
4:      current broadcast data object =  $p_0$  (target object)
5:      prefetch a data object from  $p_{-max}$  to  $p_{max}$ 
6:    }
7:    else
8:      wait until the desired data comes from the server
9:    }
10:  doze mode

```

4 Performance Evaluation

To the best of our knowledge, the scheme which broadcasts LDD without indexing has never been studied so far. Therefore, compare the performance with the other index scheme is unsuitable. Instead, in this paper, we evaluate the

performance with various kinds of parameters setting such as client's speed, size of the service area, and the distributions of the data objects.

We assume that the broadcast data objects are static such as restaurants, hospitals, and hotels. We use a system model similar to that described in [8][9]. The distance can be computed using the Euclidian distance between the two points p and q by $\text{dist}(p,q)$. The whole geometric service area is divided into groups of MSS. In this paper, two datasets are used in the experiments(see Fig 2(a)). The first data set D1 contains data objects randomly distributed in a square Euclidian space. The second data set D2 contains the data objects of hospitals in the Southern California area, which is extracted from the data set at [10]. The table 1 shows the default parameter settings used in the simulation.

Table 1. Simulation Parameters.

Parameters	Description	Setting
<i>ServiceArea</i>	Service area	1000(km)*1000(km)
<i>GroupServiceArea</i>	% of service area	30-100
<i>NoObj</i>	No. data objects	10-1000
<i>SizeObj</i>	Size of data object	1024 bytes
<i>BroadBand</i>	Broadcast bandwidth	144kbps
<i>No_Client</i>	No. of clients	0-90
<i>MinSpeed</i>	Minimum moving speed of the client	10
<i>MaxSpeed</i>	Maximum moving speed of the client	90
<i>size_W_q</i>	Size of W_q	0-5
<i>size_W_p</i>	size of W_p	0-5
<i>NoPeriod</i>	No. of broadcast period	50-100
<i>Size_max_OBC</i>	Size of max_OBC	longer than 900m

4.1 Latency

In this section, we evaluate the access latencies for various parameter settings such as the client's speed, the size of the service area, and the number of clients. In this paper, we present the Object Boundary Circle(OBC) which represents the distance between the objects as shown in Fig 2(b). The radius of circle represents the distance between objects, and a circle which has the longest radius is selected as a hot data object such as c and d in Fig 2(b). The server broadcasts data objects with different frequency such as hot and cold data objects[4].

Effect of the size of the service area. In this section, we study the effect of the size of the service area according to the client's speed. We vary the service coverage area from 5% to 100% of the whole geographic area. The query arrival time is decreased as the size of the service area decreases since the size of the entire broadcast is reduced. However, the query arrival time is significantly increased when the client's speed increases and exceeds the service coverage area as shown Fig 3(a). In this case, the client's cached data objects become invalid and the client has to tune its broadcast channel again.

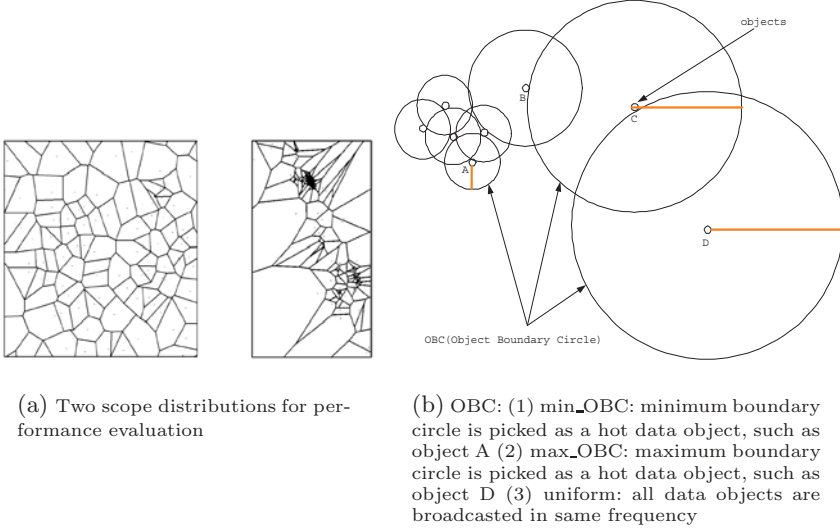


Fig. 2. Scope distributions and OBC.

Effect of the client's speed. In this section, we study the effect of the client's speed. First, we vary the client's speed from 5 to 50 in D1. When the client's speed is the lowest, a broadcast size of 10% is the best. However, as the client's speed increases, its performance is degraded in comparison with that of the other clients, whose speed exceeds the service coverage area, as shown in Fig 3(b). Second, we study the performance for different parameters such as the min_obc, max_obc and uniform(see Fig 2(b)) in D2. In this experiment, we assume that the clients are uniformly distributed in the map. Fig 3(c) shows the result as the client's speed increases from 5 to 50. Fig 3(d) shows the result as the number of clients is increased from 15 to 90.

Effect of the distributions of data objects and the clients' location.

In this section, we study the effect of the distributions of the data objects and the clients' location. First, we assume that the clients are crowded in a specific region such as downtown. Those data objects which are located in such a region are selected as hot data objects. In this experiment, we evaluate the performance in relation to four different parameters as follows:

- *uniform_100%*: The server broadcasts data objects with the same frequency such as flat broadcast in[4] and the service coverage area is the whole geographic area.
- *hot_100%*: The server broadcasts data objects with different frequencies, such as those corresponding to hot and cold data objects and the service coverage area is the whole geographic area.

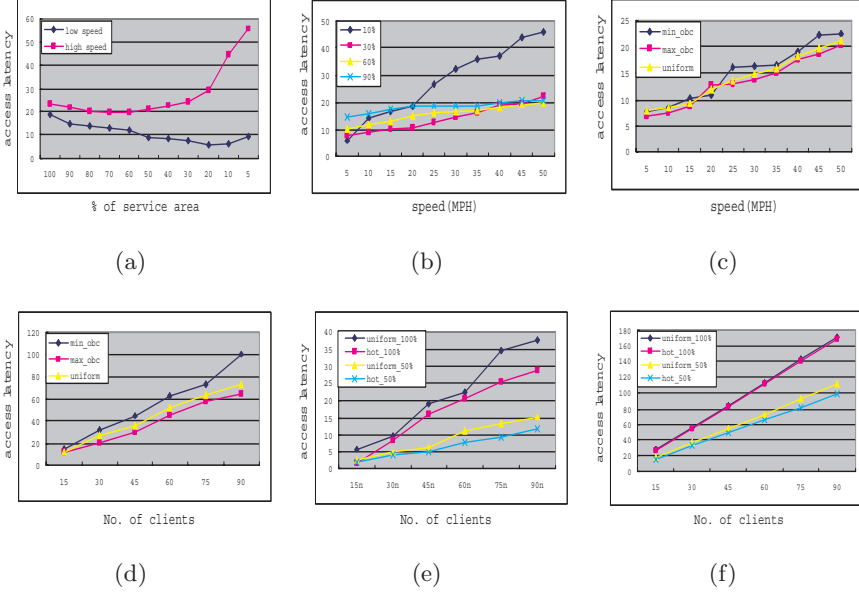


Fig. 3. Access latency.

- *uniform_50%*: The server broadcasts data objects with the same frequency, and the service coverage area is set to 50% of the whole geographic area.
- *hot_50%*: The server broadcasts data objects with different frequencies, such as those corresponding to hot and cold data objects and the service coverage area is set to 50% of the whole geographic area.

Fig 3(e) shows the result as the number of clients is increased from 15 to 90 in D1. As shown in this figure, *hot_50%* outperforms compare to others as the number of clients is increased. Second, we assume that the clients are uniformly distributed in D2. Fig 3(f) shows the result as the number of clients is increased from 15 to 90. As shown in figure, in this case, the broadcast hot data object does not affect the query response time since the clients are uniformly distributed in the map. However, the size of the service area affect the query response time.

4.2 Cache Hit Ratio

This section evaluates the cache hit ratio for various parameters settings such as the size of the w_p , the client's speed and the size of service area. First, we vary the client's speed from 10 to 50 in D2. As shown in Fig 4(a), the number of cache hits decreases as the client's speed is increased. The broadcast hot data object does not affect the client's cache hit ratio. In this case, *uniform_100%* outperforms the *uniform_50%* since clients discard the cached data object if they move to the other service area. Second, we vary the client's speed from 10 to 50 in D1. As shown in Fig 4(b), the number of cache hits decreases as the

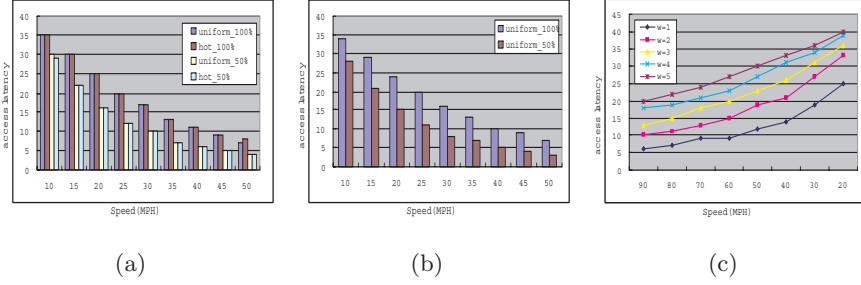


Fig. 4. Cache hit ratio.

client's speed is increased. Third, we vary the value of w_p from 1 to 5 in D1. As shown in Fig 4(c), the number of cache hits increases as the client's speed is decreased and the size of w_p increases.

5 Conclusion

In this paper, we have studied the broadcasting and prefetching schemes for LDIS. For the purpose of broadcasting in LDIS, we present the BBS and prefetching methods. The BBS method attempts to reduce the access and tuning times for the client. Furthermore, the proposed prefetching method and OBC can also reduce the query response time and tuning time respectively. With the proposed schemes, the client can perform the k -NN query processing without having to tune the broadcast channel, if the desired data objects have already been prefetched into the cache. The proposed schemes were investigated in relation to various environmental variables such as the distributions of the data objects, the average speed of the client and the size of the service area. In this paper, we did not consider the moving data objects in LDIS. Hence, we are planning to extend this study to the case of a moving object database. Finally, we are also planning to investigate the cache replacement scheme for future work.

References

1. Dik Lun Lee, Jianliang Xu, and Baihua Zheng, "Data Management in Location-Dependent Information Services," *IEEE Pervasive Computing*, 1(3), 2002.
2. T. Imielinski, S. Viswanathan, and B.R.Badrinath, "Energy efficient indexing on air," *ACM SIGMOD international conference on Management of data*, pp. 25-36, May 1994.
3. Dik Lun Lee, Jianliang Xu, and Baihua Zheng, "Data Management in Location-Dependent Information Services," *IEEE Pervasive computing*, Vol.1, No.3, 2002.
4. Swarup Acharya and Michael Franklin, "Broadcast Disks: Data Management for asymmetric communication environments," *ACM SIGMOD*, pp. 199-210, 1995.

5. W.-C. Lee and D. L. Lee, "Using signature techniques for information filtering in wireless and mobile environments," *J. Distributed and Parallel Databases (DPDB)*, 4(3):205.227, Jul. 1996.
6. Q. L. Hu, W.-C. Lee, and D. L. Lee, "A hybrid index technique for power efficient data broadcast," *J. Distributed and Parallel Databases (DPDB)*, 9(2):151. 177, Mar. 2001.
7. Jianliang Xu, Xueyan Tang and Dik Lun Lee, "Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, Vol.15, No.2, 2003.
8. Baihua Zheng, Jianliang Xu, Student Member, IEEE, and Dik L. Lee, "Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments," *IEEE TRANSACTIONS ON COMPUTERS*, VOL. 51, No.10, 2002.
9. Daniel Barbara, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," *ACM SIGMOD international conference on Management of data*, pp. 1 - 12, 1994.
10. Spatial Datasets,
<http://dias.cti.gr/~ytheod/research/datasets/spatial.html>, 2002.
11. T. Imielinski, S. Viswanathan, and B.R.Badrinath, "Data on Air: Organization and Access," *IEEE Trans. Knowledge and Data Eng*, vol. 9, no. 3, May/June 1997.

A Geocoding Method for Natural Route Descriptions Using Sidewalk Network Databases

Kouzou Noaki and Masatoshi Arikawa

Center for Spatial Information Science, the University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8904 Japan
{noaki, arikawa}@csis.u-tokyo.ac.jp

Abstract. A large amount of data including HTML documents and Internet mails has been distributed over the Internet. Most of the data on the Internet include geo-referenced descriptions. We have studied a method of converting such descriptions like addresses into their corresponding coordinates, that is, tuples of longitude and latitude. The process of converting descriptions into coordinates is called geocoding. In this paper, we focus on natural route descriptions as a new type of target to geocode. We first explain a core schema of sidewalk network databases on the basis of a characteristic of natural route descriptions, and then propose Formal Route Statement (FRS) to represent and process natural route descriptions by means of a computer. Also, we present our prototype system to geocode natural route descriptions using sidewalk network databases based on our proposed framework.

1 Introduction

We have been able to use the information space connected by networks since the birth of the Internet. However, the Internet has a problem that it is difficult for us to deal with information corresponding to locations in the real world, although we are free from barriers of time and space. On the other hand, we can acquire stable and inexpensive location data by a GPS sensor as if we can know time by a watch. Location based services in which we can obtain information based on locations are expected to be developed in the years ahead. We define *spatial content* as multimedia content with location information. We especially focus on text content with location information as part of challenges for advanced use of multimedia content.

Presently, a large amount of texts including word processing documents, HTML documents and email messages has been distributed over the Internet. Most data in our computers and on the Internet includes geo-referenced descriptions (Fig.1). When geo-referenced descriptions are converted into machine readable data, a lot of our documents will be accessed by the key of positions in the real world [1].

A method of converting such descriptions like addresses in Japanese into their corresponding coordinates, that is, tuples of longitude and latitude has been studied. The process of converting geo-referenced descriptions into coordinates is called geocoding. Geocoding for addresses was difficult in Japan because of old and complex structures of cities, but the emergence of large scale house feature databases has changed the situation of geocoding in Japan [2].

In this paper, we introduce a more advanced method of geocoding for natural route descriptions in documents using daily local expressions. Natural route descriptions are usually described in casual expressions, but not in regular ones like address. Thus, it has been considered practically impossible to geocode natural route descriptions. However, sidewalk network databases, which are one of large scale house feature databases, have been available since last year, 2003. The databases can be expected to enable more advanced geocoding for larger scale natural language expressions.

In Section 2, we explain a structure of natural route descriptions in Japanese and core schema of sidewalk network databases on the basis of a characteristic of natural route descriptions. In Section 3, we propose Formal Route Statement (FRS) to represent and process route descriptions in natural language by means of a computer. A prototype system which has been developed based on our proposed framework is introduced in Section 4 and we state conclusion and future work in Section 5.

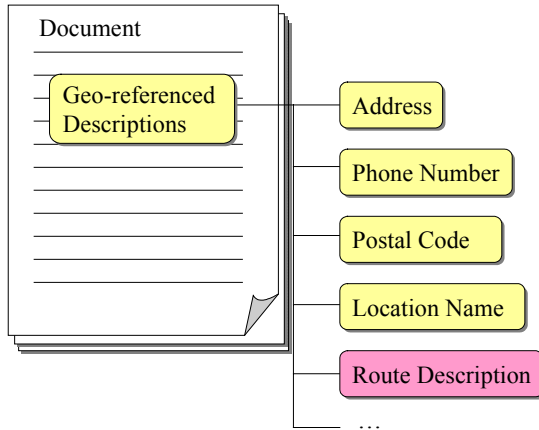


Fig. 1. Geo-referenced descriptions in documents.

2 Natural Route Descriptions and Sidewalk Network Databases

In this section, we explain a structure and characteristics of route descriptions in Japanese and then introduce the data which is required for geocoding.

2.1 Structure of Route Descriptions in Japanese

Figure 2 shows the structure of natural route descriptions which means “you exit from JR Shibuya Station east exit and go up Miyamasu Slope Street for 5 minutes, and it is on your left” in Japanese. “JR Shibuya Station east exit” is an exit of the station “JR Shibuya” and “Miyamasu Slope Street” indicates the slope street which is located on the east side of the station. The names of spatial entities correspond to *nodes* and the other words than the names are *links* between the nodes in Fig. 2. This figure shows us that a natural route description in Japanese can be divided into two components.

The first component is the names of spatial entities, and the other component is the expressions of spatial relationships between spatial entities. Constructing a natural route description can be compared to sticking of pins and stretching of a thread between pins on a map (Fig.2).



Fig. 2. Image of relations between a natural route description and spatial objects stored in spatial databases.

We define *spatial objects* as data units which correspond to entities in the world. In this paper, a focus of this study is placed on natural route descriptions used by walkers in urban cities. We especially call the spatial objects in natural route descriptions *spatial anchors*. Figure 3 shows pictures of spatial anchors in the real world. In Table 1, we collect typical spatial anchors in Shibuya, which is one of the big cities in Tokyo, Japan. These spatial anchors can be used as start points, passage points and end points of routes in natural route descriptions. According to their role in the city or



Fig. 3. Pictures of spatial anchors in Shibuya. The left picture is a scramble junction, the middle one is a slope street and the right one is a building.

characteristics of geographic features, spatial anchors can be categorized into classes such as “exit from station” or “slope street”. Each class has its own characteristics and restrictions [3]. For example, we can write “go *up* a slope street” or “go *down* a slope street”, but cannot write “go *into* a slope street”. In this case, the valid expressions following the names of spatial entities can be deduced from the classes as spatial relationship descriptions in natural language. Table 2 gives instances of spatial relationship descriptions.

Table 1. Examples of *spatial anchor* descriptions in Shibuya.

Spatial Anchor	
Building	タワーレコード (Tower Records), エイチエムブイ (HMV), QFRONT(QFRONT), 道玄坂上交番 (Dogenzakaue police box), 渋谷駅前交番 (Shibuya Station police box), ...
Exit from Station	JR渋谷駅東口(JR Shibuya Station East exit), 京王井の頭線渋谷駅 (Keio Inokashira Line Shibuya Station), ...
Street	明治通り (Meiji Street), 公園通り (Park Street), 井の頭通り (Inokashira Street), ...
Slope Street	宮益坂 (Miyamasu Slope Street), スペイン坂 (Spain Slope Street), 道玄坂 (Dogenzaka Slope Street), ...
Junction	渋谷スクランブル交差点 (Shibuya Scramble Junction), ...

Table 2. Examples of *spatial relationship* descriptions.

Spatial relationship	Examples
Direction	go forward, go ahead, advance turn to the right, on the right, on one's right turn to the left, on the left, on one's left
Distance	in 200 meter/in 5 minutes

2.2 Schema of Sidewalk Network Databases

Sidewalk network databases store underground walks, footbridges and cross walks for pedestrians. Sidewalk network databases are simply structured as nodes and links. A node has geometric coordinates. A link is defined as a vector between two nodes. Sidewalk network databases are provided as commercial products by Shobunsha Publications Inc. [4]. The commercial sidewalk network databases presently cover major cities in Japan.

Before the emergence of sidewalk network databases, there have been popular geographic network databases such as road networks for car navigations, railroads,

facilities networks and so on. However, the previous geographic network databases were all designed for small scale uses, not for large scale uses such as human navigations. On the other hand, sidewalk network databases for walkers are getting popular for human navigation systems since 2003 in Japan. Some services and products using sidewalk network database have already been on the market. “EZnaviwalk” provided by KDDI au is one of the most popular human navigation services using cell phone, GPS, and electric compass [5]. With sidewalk network databases, train timetables and airline timetables, EZnaviwalk finds the most direct, time-saving or money-saving route. This service has begun since October 2003. At present, October 2004, there are more than 100,000 users of EZnaviwalk. In addition, Shobunsha Publications Inc. is going to market digital barrier-free maps for all pedestrians including the elderly and the disabled in November 2004.

The sidewalk network database used in our research is simply structured as nodes and links. A node has not only geometric attributes like coordinates but non-geometrical attributes like name of spatial entity, its class and related url. A link has a distance and an angle from its start node to its end node. Users can extend the integrated database by entering additional nodes and links or inputting text data. In particular, “junction”, “street” or “slope street” is shaped with the multiple nodes and links. For example, at least four nodes are needed to shape a junction (Fig.4.). Figure 5 shows the core schema of sidewalk network databases. A node has attributes, that is, *id* (the identifier of a node), *name* (a spatial entity’s name which corresponds to the node), *coordinates* (tuples of longitude and latitude), *in* (a name of a street or an area both of which are constructed of multiple nodes and links), *class* (a class of spatial object), *incoming link* (*id* of an incoming link), *outgoing link* (*id* of an outgoing link) and *poi* (additional information concerning the point of interest except for *name*, *class* and *url*). A link has its *id* (the identifier of the link), *start_node* (*id* of the node from which the link starts), *end_node* (*id* of the node at which the link arrives), *direction* (of the link in degree) and *distance* (of the link in meter).

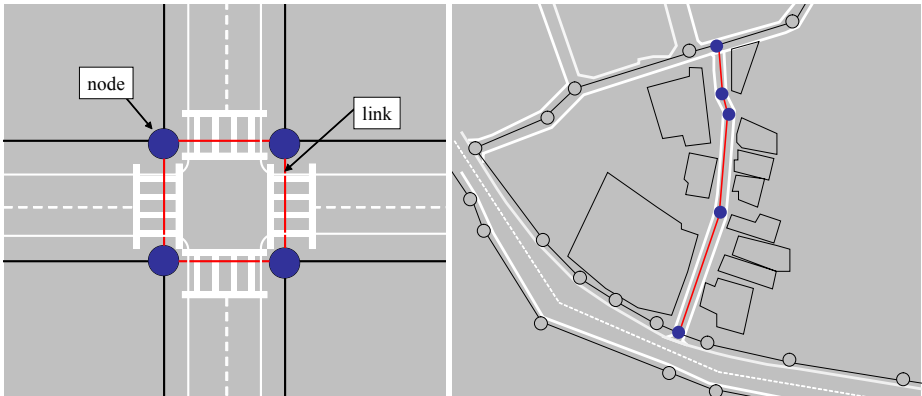


Fig. 4. Examples of nodes and links stored in sidewalk network databases. The left figure shows a junction and the right one shows a slope street.

<i>Sidewalk network DB :</i>	<i>(Nodes, Links)</i>
<i>Nodes :</i>	a set of <i>node</i>
<i>Links :</i>	a set of <i>link</i>
<i>node :</i>	<i>(id, name, coordinates, in, class, incoming_link, outgoing_link, poi)</i>
<i>link :</i>	<i>(id, start_node, end_node, direction, distance)</i>
<i>node.id :</i>	<i>id of the node</i>
<i>node.name :</i>	<i>name of the node</i>
<i>node.coordinate :</i>	<i>coordinates (longitude, latitude)</i>
<i>node.in :</i>	<i>name of group</i>
<i>node.class :</i>	<i>class of spatial object</i>
<i>node.incoming_link :</i>	<i>id of the incoming link</i>
<i>node.outgoing_link :</i>	<i>id of the outgoing link</i>
<i>node.poi :</i>	<i>information for the point of interest</i>
<i>link.id :</i>	<i>id of the link</i>
<i>link.start_node :</i>	<i>id of the start node</i>
<i>link.end_node :</i>	<i>id of the end node</i>
<i>link.direction :</i>	<i>its direction in degree</i>
<i>link.distance :</i>	<i>its distance in meter</i>

Fig. 5. Schema of the sidewalk network databases used in our research.

3 Formal Route Statements (FRS)

We assume that most of natural route descriptions consist of nodes which indicate start, passage or end points and links between them. Moreover, the end point, in other words the destination point, is often omitted in natural route descriptions, especially in Japanese, because of being included somewhere in documents.

Formal statements are necessary for computers to indirectly deal with natural route descriptions. On the assumption that all of natural route descriptions can be expressed with nodes and links, we propose *Formal Route Statement (FRS)* to represent and process natural route descriptions. FRS also works as a query language for the sidewalk network databases. Figure 6 shows the grammar of FRS. Generalization

<i>FRS ::=</i>	<i>node_desc(0):(link_desc(i):node_desc(i+1))*</i> <i>[i={0,...,n}];</i>
<i>Node_desc(i) ::=</i>	<i>node(i).node_attribute_list</i>
<i>node_attribute_list ::=</i>	<i>none node_attribute_value (&node_attribute_value)*</i>
<i>node_attribute_value ::=</i>	<i>node_attribute = value</i>
<i>node_attribute ::=</i>	<i>id name coordinate in class status</i>
<i>value ::=</i>	<i>numerical_value string_value url status_values </i> <i>connect_values</i>
<i>status_values ::=</i>	<i>start end via</i>
<i>connect_values ::=</i>	<i>straight right left</i>
<i>link_desc(i) ::=</i>	<i>link(i).link_attribute_list</i>
<i>link_attribute_list ::=</i>	<i>none link_attribute_value (&link_attribute_value)*</i>
<i>link_attribute_value ::=</i>	<i>link_attribute = value</i>
<i>link_attribute ::=</i>	<i>id start_node(id) end_node(id) direction connect </i> <i>distance</i>

Fig. 6. Grammar of Formal Route Statement (FRS).

tables are indispensable to converting various casual descriptions into regular ones, one of which is FRS (Table 3). A use case of the generalization tables is to make an instance of the spatial relationship as a value of the attribute “link.connect” in Fig. 6. The attribute “link.connect” plays an important role to find a node when a name of the next node is omitted.

Table 3. Example of a generalization table for descriptions of spatial relationship and values of the attribute “link.connect”.

Specialized descriptions for spatial relationships	Generalized descriptions for spatial relationship (values of link.connect)
go forward, go ahead, advance	Straight
turn to the right, on the right, on one's right	Right
turn to the left, on the left, on one's left	Left

Figure 7 shows an example that “JR渋谷駅東口より、宮益坂をのぼって約5分、左手” (means “you exit from JR Shibuya Station east exit and go up Miyamasu Slope Street for 5 minutes, and it is on your left”.) is converted into the following FRS.

```

node_desc(0) = “JR 渋谷駅東口 (JR Shibuya Station east exit)”
link_desc(0) = “より, (from)”
node_desc(1) = “宮益坂 (Miyamasu Slope Street)”
link_desc(1) = “をのぼって約5分, (go up for 5 minutes)”
node_desc(2) = “”
link_desc(2) = “左手 (on your left)”
node_desc(3) = “”
FRS = node_desc(0):link_desc(0):node_desc(1):link_desc(1):node_desc(2)
      :link_desc(2):node_desc(3)

```

Fig. 7. An example of FRS.

4 Prototype System

Our proposed framework in Section 2 and 3 has been verified through developing a prototype system. Figure 8 shows the user interface of the prototype system.

4.1 Overview of Prototype System

We have developed a prototype system which processes a route description in Japanese and then visualize it as a polyline on the map using sidewalk network databases. We explain each component in the user interface (Fig.8) as follows;

(A) Menu button.

Users can change the operation mode by the menu buttons. Main functions are (1) loading and saving network data and (2) adding, erasing and moving both nodes and links. Furthermore, we select functions of referring to node information (for example, a name of a spatial entity) and of changing the mode of filling in entry forms.

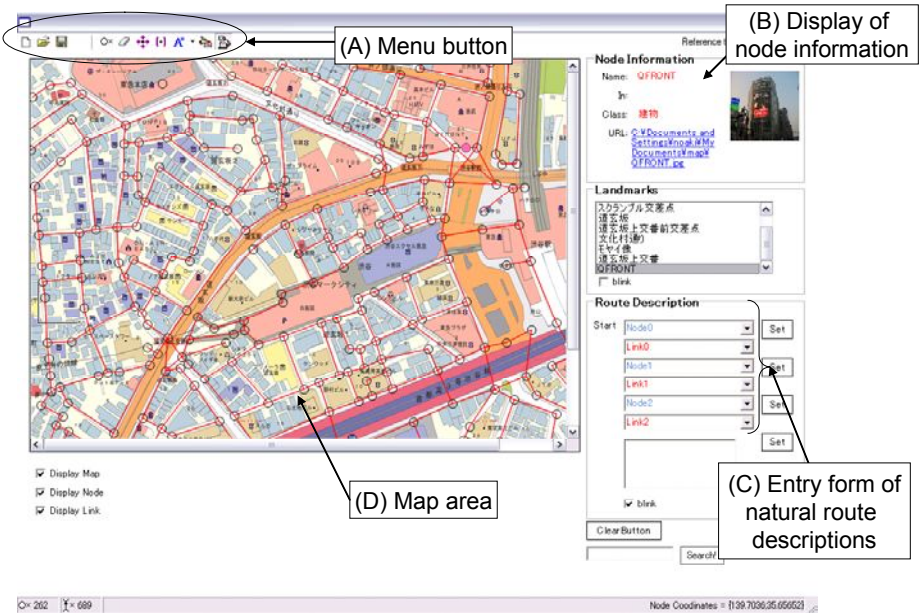


Fig. 8. Graphical user interface of the prototype system.

- (B) Display of node information.
This area allows users to see the name, class and picture image of a spatial anchor on the map area.
- (C) Entry form of natural route descriptions.
These forms allow users to enter natural route descriptions, and then convert the descriptions into FRS as queries for sidewalk network databases.



Fig. 9. Dialog box of entry form for node information.

- (D) Map area.
Sidewalk network databases and a map image in the selected area are overlapped and visualized. Moreover, a route is also visualized on the sidewalk network as a result of geocoding a natural route description. Figure 9 shows a dialog box for an

entry form of node information, which allows users to add their collecting POI data (including a name and a class for a spatial entity). This window appears when users push the entry button in menu buttons and click a node on map area.

4.2 Experimental Demonstration

We simply explain an algorithm of geocoding for natural route description using sidewalk network databases as follows:

- A) **Spatial anchor point match:** The names of spatial anchors in a natural route description are matched with sidewalk network databases through queries. The results of the queries are a set of nodes corresponding to the spatial anchors.
- B) **Path match:** This step decides the sequence order of locations in the route using the rule which minimizes each of the moving costs between neighbor locations.
- C) **Dealing with insufficient descriptions:** If a name of spatial entity is omitted or not specified explicitly, the name is deduced from descriptions of spatial relationships between spatial entities.

Figure 11 shows behavior of processing Japanese route descriptions meaning that “you exit from *JR Shibuya Station Hachiko exit* and go up *Dogen Slope Street*, and then turn right at *the junction in front of the police box on the top of Dogen Slope Street*”(Figure 10). The *Italic parts* in the above route description are names of spatial anchors.

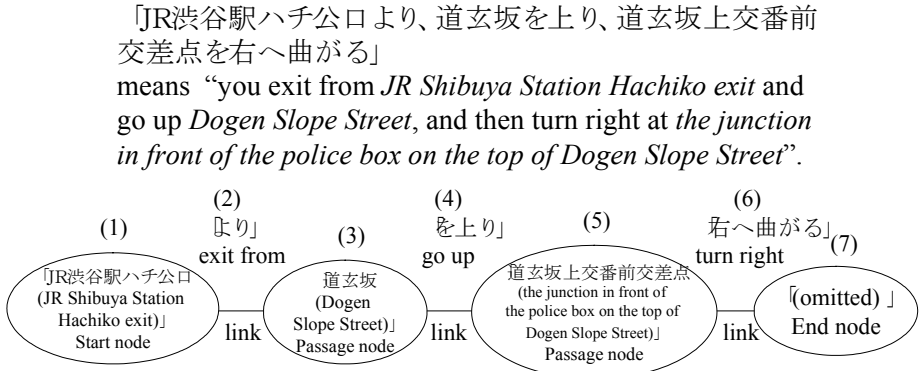


Fig. 10. Example of a natural route description and its corresponding FRS.

A natural route description is converted into FRS. The route can be visualized on a map because each node has the data of longitude and latitude. In the above example, the number of the result route satisfying a FRS can be only one. However, in general, there are multiple result routes and they should be ranked with the levels of quality for practical applications. For example, a time distance “5 minutes” can be translated into a distance “400m”, or multiple distances “300 or 400 or 500m”, or a range of distance “300m-500m” because a speed of walking depends on the environment.

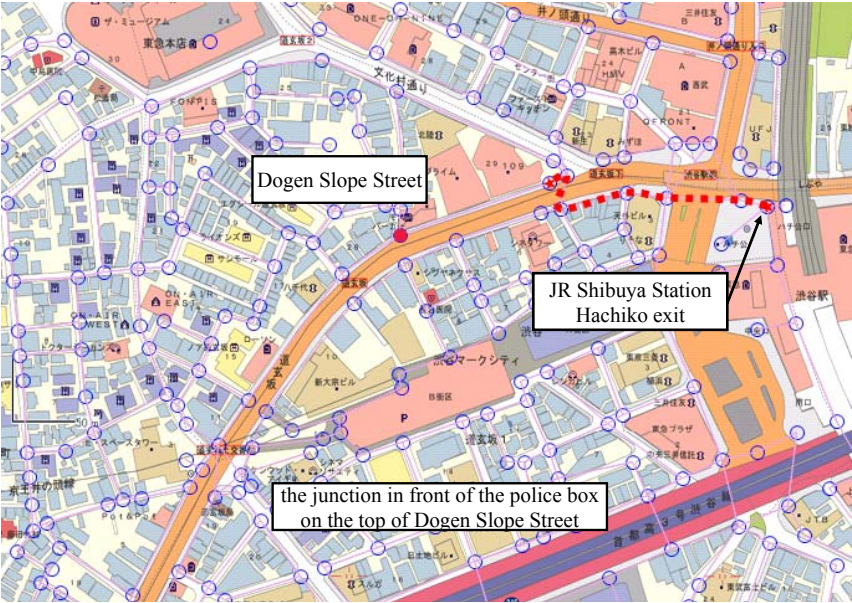


Fig. 11(a). Our prototype system searches the shortest path (the dashed line) from the node matched “JR Shibuya Station Hachiko exit” to the nearest one of the nodes making up “Dogen Slope Street”. This figure shows the result of processing (1)+(2)+(3) which are parts of the natural route description in Fig.10.

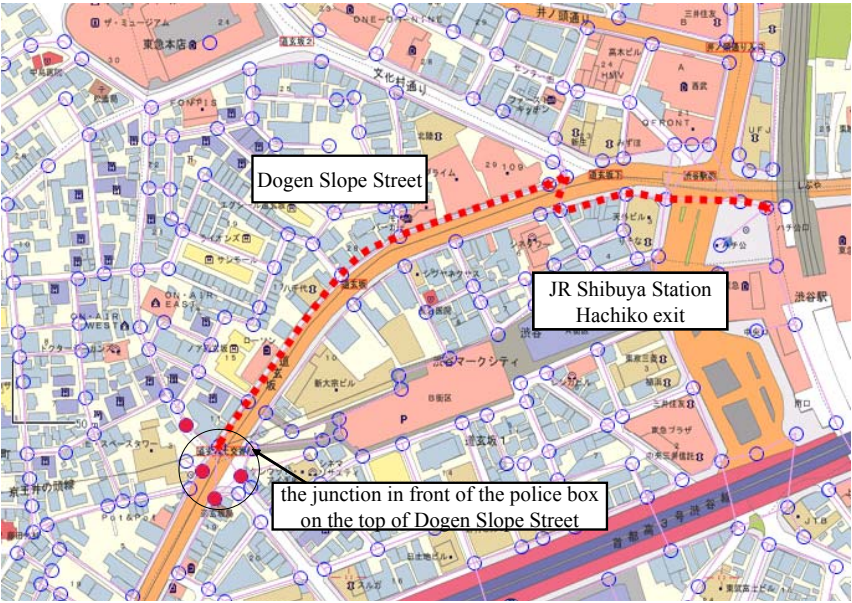


Fig. 11(b). It searches the shortest path to the nearest one of nodes of “the junction in front of the police box on the top of Dogen Slope Street”. This figure shows the result of processing (1)+(2)+(3)+(4)+(5) which are parts of the natural route description in Fig.10.

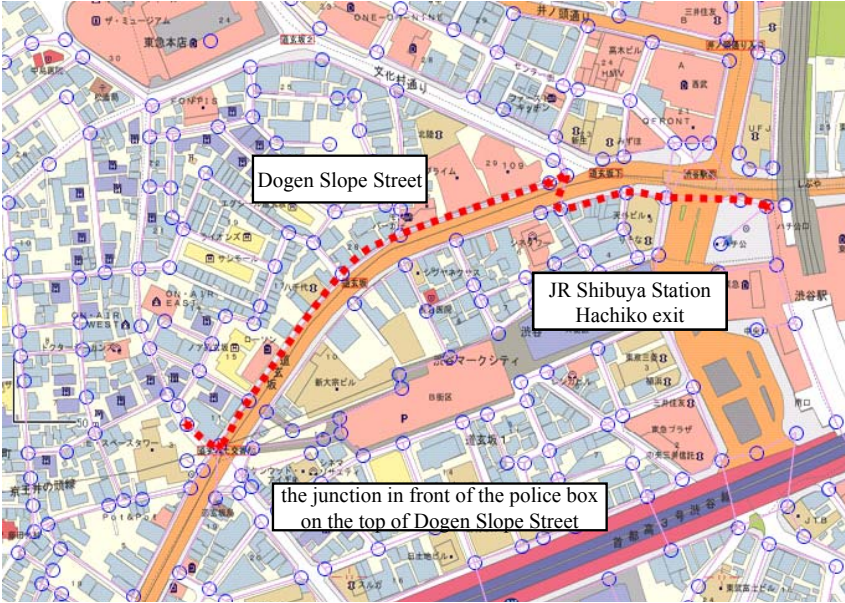


Fig. 11(c). The end node is deduced from the description of “turn right” and the direction of its incoming link. This figure shows the result of processing (1)+(2)+(3)+(4)+(5)+(6)+(7) which are parts of natural route description in Fig.10.

5 Conclusion and Future Work

In this paper, we proposed a basic framework to geocode natural route descriptions by means of sidewalk network databases. The prototype system has been developed to verify our proposed framework. The primary function of the prototype system is a naïve database management system for sidewalk network databases. Also, its secondary functions include both FRS validator and visualizer. Valid FRS means that the topology composed of its spatial anchors and its spatial relationship expressions can be matched in sidewalk databases.

The framework presented in this paper is only a part of our framework (Fig.12) and we are going to develop FRS generator which can extract natural route descriptions from various documents and then convert the natural route descriptions into FRS’s. We have to construct a series of algorithms as following;

- (1) Extracting natural route description from document data
- (2) Separating spatial anchor’s names and spatial relationship expressions in a natural route description
- (3) Generalizing spatial anchor’s names and spatial relationship expressions
- (4) Generating FRS (which is grammatically correct)

In the above (1) and (2), there are already significant achievements in the fields of natural language processing [6]. In the above (3), we plan to realize the functions to complement the incomplete name of a spatial anchor and correct inappropriate natural route descriptions using sophisticated geographic thesaurus and generalization rules for names.

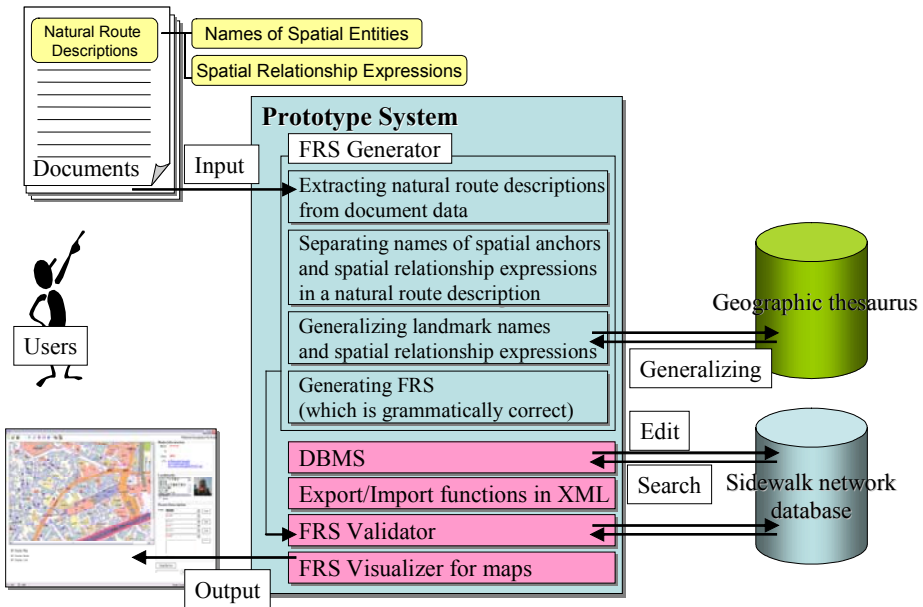


Fig. 12. Configuration of the prototype system.

A part of DataBase Management System (DBMS) in this prototype system is a minimum formation. We will extend functions and database. For example, descriptions after names of spatial anchors can be anticipated by using the knowledge of not only the class of spatial anchors but also geographic features stored in the sidewalk network databases. In walking starting from a lowest place, it is clear that we cannot go down there. In addition, multiple solutions, which are possible routes to a destination, should be ranked by some criteria of quality of the geocoding results.

This study proposed one of the methods of geocoding to convert geo-reference text data in digital documents into tuples of longitude and latitude. As a result of establishment of these methods, we will be able to manage many documents as points and polylines on a map with advanced spatial relationships.

Acknowledgments

We would like to thank Dr. Takeshi Sagara and all members of Spatial Media Fusion Project for their valuable comments to our research. This work was supported in part by a Grant-in-Aid for Scientific Research on the Priority Area "Informatics Studies for the Foundation of IT Evolution" by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Kaoru Hiramatsu: Studies for Web information retrieval using geo-reference (in Japanese), Doctoral dissertation (2002).

2. Takeshi Sagara: Studies for advanced practical use of non-structured and semi-structured data (in Japanese), Doctorial dissertation (2003).
3. Annette Herskovits: Language and spatial cognition, Cambridge University Press, Melbourne (1986).
4. Shobunsha Publications Inc., <<http://www.mapple.co.jp/>>
5. EZnaviwalk, KDDI au, <<http://www.au.kddi.com/>>
6. Makoto Nagao, Satoshi Sato, Sadao Kurahashi, Tatsuhiko Tsunoda: Natural language processing (in Japanese), Makoto Nagao (edit), Iwanami Shoten, Tokyo (1996).

Location-Based Tour Guide System Using Mobile GIS and Web Crawling

Jong-Woo Kim¹, Chang-Soo Kim¹, Arvind Gautam², and Yugyung Lee²

¹ PuKyong National University, Interdisciplinary Program of Information Security,
599-1, Daeyeon3-Dong, Nam-Gu, Pusan 608-737, Republic of Korea
jwkim73@mail11.pknu.ac.kr, cskim@pknu.ac.kr

² University of Missouri at Kansas City, School of Computing and Engineering,
Kansas City, MO 64110, USA
{asg3hb, leeyu}@umkc.edu

Abstract. Location-Based Service(LBS) is a wireless application service that uses geographic information to serve a mobile user. Recent research on the LBS focuses on context sensitive computing and visualization. Location is the core context in context sensitive LBS and the use of the map is the most efficient way to visualize the geographic information. Many existing researches, however, has limited capability to provide dynamic map services and realistic information acquired from the Web. The goals of our research are to fully access geographical information and incorporate it with the LBS services. For the purpose, it is important to extract semantically relevant geographical information from the Web and efficiently present it to mobile users. In this paper, we present an effective LBS approach based on advance Mobile GIS and Web technologies. We design and implement the prototype of the Tour Guide System as a motivating application of LBS.

1 Introduction

In recent years there has been a growing interest in using Geographical Information Systems(GIS) and wireless devices for geospatial services. Among a few exciting geospatial applications that governments and industry have developed, Location-Based Service(LBS) is one of the most emerging GIS applications. The LBS concentrates on providing a wireless application service to mobile users derived from geographic information and the location information provided by mobile devices. LBS, therefore, is the capability to find the geographical location of the mobile device and provide services based on this location information. As an example, when asking for information about the closest hotels, the user would be provided with the relevant information about hotels closest to the user's current location(determined by the LBS system). Several other LBS applications have been developed within a certain context like emergencies, route finding, etc.

Tour Guide systems have been recognized as a useful LBS application. Tourists may be unaware of visited areas and seek the location sensitive services

for the most frequently visited areas. Such services of interest may include cities' attraction point information, weather services, emergency services, roadmap services, currency changes, etc. Recent work in the area of the Tour Guide system provides not only the location of such services but also additional information about the place, such as pictures of the establishment, a detailed description of the service, etc. The increasing requirements to handle more information on mobile devices are putting more demands on context sensitive computing and visualization than ever. In the context sensitive tour services, location is the core context in the tour domain. Thus, it would require to access dynamic and evolving information depending upon user's location-relevant contexts such as his/her current location, surrounding entities and services, mobility constraints, and traffic. Visualization is another important component for the Tour Guide system and it needs to build an efficient way to display user's location and relevant information of the user selected geographic location.

In this paper, we focus on two issues: visualization and location based service. A static map used in the most of existing systems hardly provides context sensitive information to mobile users. On the other hand, a digital map represented as vector data is flexible to provide dynamic services such as presenting various views through scaling and spatial level of details. There are some serious challenges for providing the capability to use digital maps in a mobile device.

Extracting information on location-based services is also difficult since it is not easy to get such updated information. Even if the Web provides such information, extracting LBS information from the Web is also difficult. It is mainly the data over the Web is not structured. In addition, the great challenges are associated with the dynamic features of the Web such as periodically update services, continuously generated new services and heterogeneous service representation.

Our intent to resolve these issues is to improve our previous work on efficient Mobile GIS model[4] and extracting information over the Web[7]. We developed a Tour Guide system to demonstrate the location based tour services/information crawled from the Web. The system provides the following functionality: (1) discover the location specific information of user-preferred attractions near his/her current location from the Web, (2) display the locations of these attractions on a digital map, (3) provide location based Web services for Tour Guide applications on mobile devices.

This paper is organized in the following manner: In Section 2, we present related work, followed by overview of the location based application, the Tour Guide system in Section 3. In Section 4 and Section 5, we describe the architecture and implementation of the prototype system. In Section 6, we end with a concluding statement and future work.

2 Related Work

In order to provide geographic information in mobile environments, various studies are being done on Mobile GIS and Location Based Services(LBS). At

present the key subjects under the GIS technology umbrella are - Internet GIS¹ and 3D GIS². Organizations that are involved in fixing the standards are - OpenGIS³(OpenGIS GML, OpenGIS, OLE), Open Location based Services⁴(OpenLS), Initiative,ISO/TC211⁵, LIF⁶(Location Interoperability Forum), MAGIC Services Forum, etc. Stockus, et al.[9] worked on LBS and the integration of GPS data into an embedded Internet GIS. In [10], the framework and efficient data exchange protocol were proposed to use a large amount of geospatial data in restricted mobile environments.

Kushmerick et al.[5] introduced the concept of creating specialized wrappers covering specific sources of information over Internet. This method uses Inductive Learning approach to automate the creation of Wrappers. RAPIER [2] takes pairs of sample documents and filled templates and induces pattern-match rules that directly extract fillers for the slots in the template. RAPIER employs a bottom-up learning algorithm which incorporates techniques from several inductive logic programming systems and acquires unbounded patterns that include constraints on the words, part-of-speech tags, and semantic classes present in the filler and the surrounding text. There have been a number of researches to provide context sensitive Tour Guide information through Web-based system. The recent researches focus at context sensitive computing and visualization.

Abowd et al.[1] have developed Cyberguide, which is handheld electronic tourist guide system that supplies the user with information based on user's location. Initially Cyberguide was developed for indoor tours at the Gvu Center with Active Badge system. The system was extended to operate outdoors with GPS. This system provides the following components: a map component for cartographer service, information component for librarian service, positioning component for navigator service and communication component for Messenger service. Their experimentation with bitmap and vector-based map encountered problems of using vector-based maps. As they used workstation-based GIS mechanism, the system required high bandwidth downstream wireless connectivity to a wireless mobile client.

Davis et al.[3] has an ongoing project GUIDE to investigate electronic tourist guides in a practical real-world environment. They have been building and testing different versions of electronic tourist guides for the city of Lancaster over the past few years. Their current approach uses wireless communication on a pen based tablet computer. They intend to consider not only the location but also the visitor's interests, the city's attractions, mobility constraints, available time, weather and cost as context for their context sensitive Tour Guide system.

Simcock et al.[8] focus on software support for location based applications. They are not just interested in the location but also other elements, attractions

¹ http://www.itc.nl/education/programme_levels/module_descriptions/ELEC00709.asp

² <http://www.davidrumsey.com/GIS/3d.htm>

³ <http://www.gis.com/software/ogc.html>

⁴ <http://www.opengeospatial.org/initiatives/?iid=73>

⁵ <http://www.isotc211.org/>

⁶ <http://www.cellular.co.za/lif.htm>

and equipment near by. Their main aim is to develop a context sensitive travel expo application. The application consists of a map mode, a guide mode, and an attraction mode. It provides map service based on bitmap-based map and tourism information service through HTML-based sound, images, and text.

Although these systems provide context sensitive Tour Guide information based on location and visualization through HTML-based content, there still remain several challenges to overcome the limitations of their approaches, image map and static information and provide efficient map service and dynamic location-based service.

3 The LBS Application – Tour Guide System

In this section we use a LBS scenario to explain the high level of our Tour Guide System. Through this scenario, we highlight how to provide service dynamic location-based services crawled from the Web and how to present dynamically generated map for mobile devices using our efficient Mobile GIS model.

The architecture of the Tour Guide System is shown in Figure 1. Our System can provide the information of user-preferred attraction points ranked by the distance from the user's present location on the map, and the user can access their Website/Web Service by clicking on it on the digital map. For example, if a user wants to find all hotels in 2 miles radius centered at his/her current location, our system shall proceed in the following manner:

1. The Mobile Client receives the current location(Longitude/Latitude) of the user through GPS.
2. The Mobile Client sends a query to the Tour Guide Server asking it to locate all hotels within the 2 miles distance from the current location of the user.
3. The Tour Guide Server crawls the websites of the hotels within the 2 miles distance from user's location.
4. The Tour Guide Server responds with the location-information and Web-URLs of the searched hotels to the Mobile Client.
5. The Mobile Client screen displays the searched hotels on the digital map by resolving the GPS coordinates of the specified locations and synching them to those of the map.
6. The Mobile Client accesses the Web Services/Web Sites to obtain additional information of the hotel selected by the user.

The primary components of our system include:

- Positioning component: Because user's location is most important context in a tour guide system, the component to determine location is necessary. To determine location, two technologies can be used: cellular location-based system and GPS(Global Positioning System). We used the GPS in our system because of its global aspect and high accuracy(approximately 10 meters). The positioning component contacts GPS to receive the current location information of user.

- Digital map service component: A map service is a simple and efficient way to present map information to the uses of the tour guide system. This component enables the users to view their own location and points of interest on the digital map. The digital map service provided is based on vector spatial data which has many advantages compared to image maps. For this component, we explore the efficient Mobile GIS model developed in our previous work[4].
- Service discovery component: This component is responsible for two tasks: (1) discovering/extracting the geo-spatial information embedded into documents over Web using our OntoGenie approach[7] (2) providing a query resolution service using an inference engine to map user’s queries to appropriate types defined in a Service ontology.
- Communication component: In our system, the major role of the client is to afford a Web service interface for a map service. Location-based information is crawled from the Web by the Server. It is therefore necessary to communicate requests/responses between the client and the server to service the location-based information. A SOAP-based communication is established between the client and server. This component provides the functionalities to create and parse the SOAP request/response message according to user requirements.
- Information component: A tourist wants information about points of interest in the surrounding area. In case of the point of interest being a hotel, the user wants to know not only the location of the hotel but also the rating of the hotel, kinds of rooms, service charge, etc. In addition, the user wants to know whether he/she can make a reservation for a selected hotel.

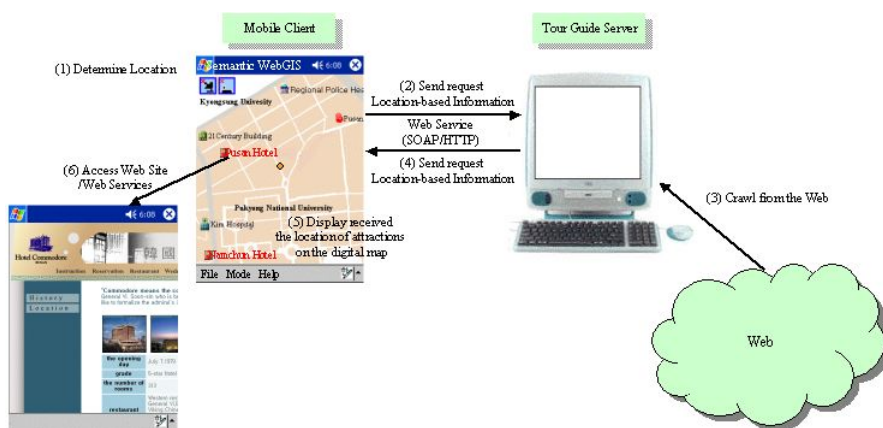


Fig. 1. Location based Tour Guide System.

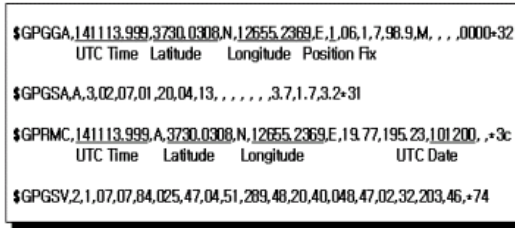
4 Realizing the Tour Guide System

In this section, we describe the implementation details of the major components in the Tour Guide System.

4.1 Positioning Component

The emerging cellular location-based systems typically provide accuracy within approximately 50 meters, and they are broadly used for assisting emergency services providers in locating callers. The GPS, the most widely deployed location technology system, is a satellite-based navigation aid originally developed by the US military. The GPS receivers obtain signals from multiple satellites and use a triangulation process to determine their physical location, which is accurate to within approximately 10 meters. In the paper, we utilize GPS to determine user's location.

The GPS receiver outputs GPS information using NMEA-0183 protocol which is defined by NMEA (National Marine Electronic Association). There are GGA, GSA, RMC and GSV messages provided within the NMEA-0183 protocol. The GGA and RMC messages include location information (Figure 2). In the paper, we get UTC Time, Latitude and Longitude from GPS information using GGA message format. However, the location information received in GPS is not used to show current location on digital map because GPS uses longitude/latitude coordinate system and digital map works on TM (in Korea)/UTM (in USA) coordinate system. Our system provides a conversion between the two coordinate systems using the principal of Gauss-Kruger Projection.



```

$GPGGA,141113.999,3730.0308,N,12655.2369,E,1,06,1,7.98,9,M,...,0000-32
      UTC Time  Latitude  Longitude  Position Fix
$GPGSAA,3.02,07,01,20,04,13,...,3.7,1.7,3.2-31
$GPRMC,141113.999,A,3730.0308,N,12655.2369,E,19.77,195.23,101.200,...,3c
      UTC Time  Latitude  Longitude  UTC Date
$GPGSV,2,1,07,07,84,025,47,04,51,289,48,20,40,048,47,02,32,203,46,+74

```

Fig. 2. NMEA-0183 messages received from GPS.

4.2 Digital Map Service Component

For the Map Service Component, we expend our previous model (Figure 3) developed for an efficient mobile GIS[4]. Standard digital map formats such as DXF, GML and Shape are relatively bulky because they tend to include many real-world spatial data. However, it is necessary to reduce map data volume so that these data could be presented on mobile devices. Typically the mobile devices are limited in low processing power, low memory, small screen size and

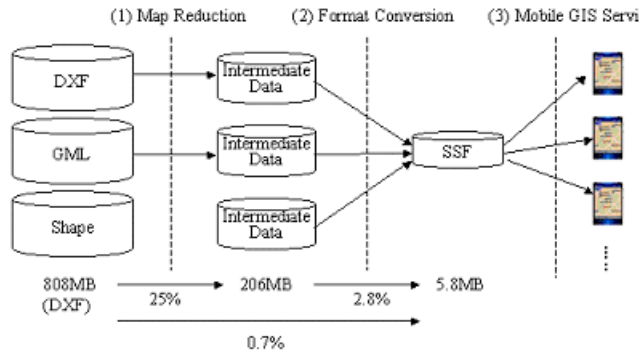


Fig. 3. The Mobile GIS Model developed in our previous work.

limited communication environments. It is a demanding need for an advanced GIS technique which can provide map reduction and format conversion services. The Map Service Component supports such service to present geographic information efficiently on mobile devices like PDA.

The objective of the mobile GIS model is to perform a task of map reduction to remove unessential geographic data. The map reduction process is composed of four steps as shown in Figure 4: (1) dividing a digital map into smaller pieces (2) generating map (3) creating polygons (4) converting geographic data into efficient data for mobile devices. At the first step, a digital map is divided into smaller parts suitable for a PDA's display size. At the map generalization step, details are selectively suppressed and an abstract model is built. The generalization is done using operations such as selection, simplification, and symbolization[4]. At the third step, polygon creation is done by combining

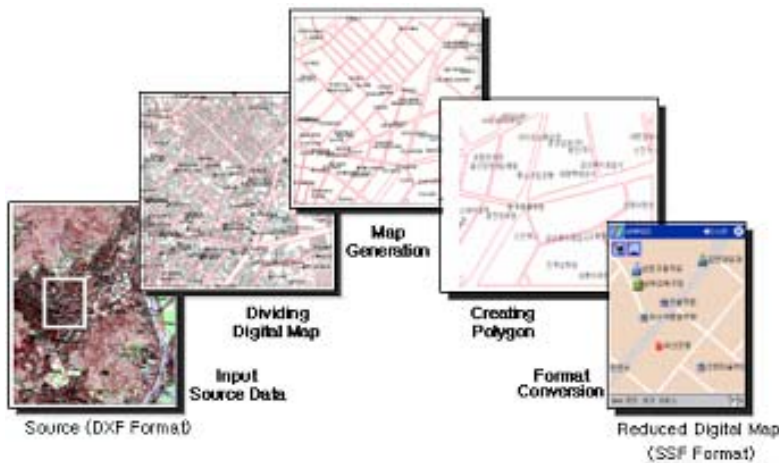


Fig. 4. Efficient Mobile GIS Model.

polylines. The polygon creation provides a clear/better visualization of the map because a polygon is a closed figure and can be filled with color. At the format conversion step, the DXF format of the map is converted into a Simple Spatial data Format (SSF) format we developed[4].

The purpose of SSF is to divide geographic coordinates into a base coordinate set and offset(Figure 5). For the base coordinate set, Xmin and Ymin in the file header are used. In order to represent the location of a spatial object, only the offset is used instead of the real coordinate set. A coordinate set generally requires 16 bytes or more of memory, while SSF requires only 8 bytes. Considering a map consisting of numerous records and coordinate sets, the use of SSF may result in a significant reduction of the storage

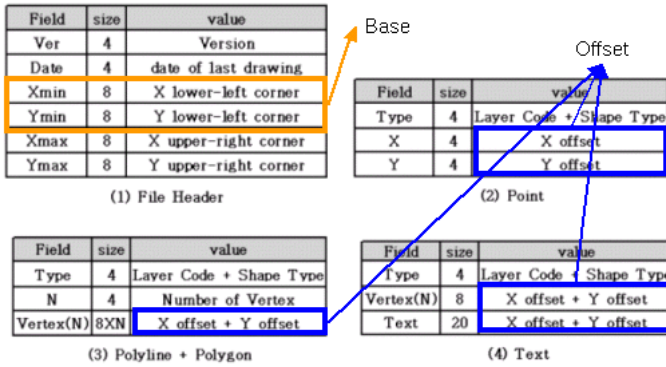


Fig. 5. Simple Spatial data Format(SSF).

Using our Mobile GIS model, the size of DXF was actually reduced as shown in Figure 3. More specifically, after the map reduction, the volume of the digital map was reduced from 808MB to 206MB. The conversion of the digital map into SSF reduced it into 5.8MB. The map size in overall was reduced from 808MB to 5.8MB. The reduced digital maps are to be stored on mobile devices such as PDA as well as on the GIS server. For the latter case, the digital maps will be served to the PDA through wireless communication between GIS server and mobile devices, and the Mobile GIS model is supposed to use during communication. In this way it was ensured that the efficient Mobile GIS model improves the communication between the server and mobile devices.

4.3 Service Discovery Component

The service discovery in our system can be achieved through Service Ontology instantiation. This means that we make ontology instantiation dynamic and drive the Web-based service extraction based on the information required for instantiating a particular service domain. For instance, while instantiating Hotel ontology in the ontology, we look for hotel service information specific to the

location over the Web and populate such concepts and properties of the Hotel ontology with related hotel service information. The use of the service ontology allows us to find the closest match to a particular request; this is due to the fact that the Service Ontology gives us the ability to inference relations between concepts.

For service information searching, we have built a Web-Service wrapper over the Google Web Crawler⁷. For information extraction from the crawled services, we extend our OntoGenie approach[7] as part of our ongoing effort to meet the needs of extracting location specific service information from the Web data and populate the Service Ontologies with the extract information. The populated Service Ontology(Figure 6) is representing the semantic location-based services specific to the user's query. Now for a given query, the Service discovery component starts from the concrete concepts in the query to broader concepts - as defined by the Service Ontology.

The Service Discovery Component also provides a query resolution services that uses a Jena⁸ based inference engine to resolve the user's queries to appropriate types through a defined Service Ontology(Figure 6). When a query is given by the user to search for a particular service, the inference engine resolves the query to a service type defined in the ontology and finds compatible services in an area close to the user's location. The address information of the services returned as result is converted to Geo-coordinates using Geo-coding Web-service⁹. This data is then used to extract the linear surface distance to these locations from the user's location through a Linear-distance lookup service¹⁰. These Service locations are then ranked by distance from the current location and by closeness of conceptual match to the query. Currently, we use the maps indicating the location of the services pulled from map services on the Web (like www.mapquest.com) or map services through Web services (such as provided at www.mappoint.com) by providing the address to be displayed on the map. These map files provided are small files (20KB) and can be easily transmitted over a wireless/non-wireless network. Additional information, such as URLs of Websites about the user's location, property values of the service extracted from crawled service Web pages, etc. is also available through this component.

4.4 Communication Component

Web Services are an emerging technology that aims at integrating applications distributed over heterogeneous environments. The success of Internet and the Web has been attributed to the standardization of protocols and development of tools and applications for Web Services. Simple Object Access Protocol (SOAP)[6] is one of the Web standards that provide specifications to realize a

⁷ <http://www.google.com/apis/>

⁸ Jena - <http://jena.sourceforge.net>

⁹ Geocoding - <http://www.geocode.com>

¹⁰ Linear distance - <http://ws.cdyne.com/psaddress/addresslookup.asmx?op=CalculateDistanceInMiles>

service based middleware over Web for distributed applications. The XML-based representation of SOAP is used as a means for transmitting service requests and responses over Web protocols - performing remote procedural calls using Web protocols like HTTP. This light protocol enables the exchange of information in a decentralized, distributed environment. The simplicity and extensibility of the SOAP enables communication between applications implemented on different platforms.

In our Tour Guide System, SOAP based Web Services are used to communicate request/response for location based information between Mobile Client and Tour Guide Server. Figure 7 shows SOAP Request/Response Message (getAddress) used in our Tour Guide System.

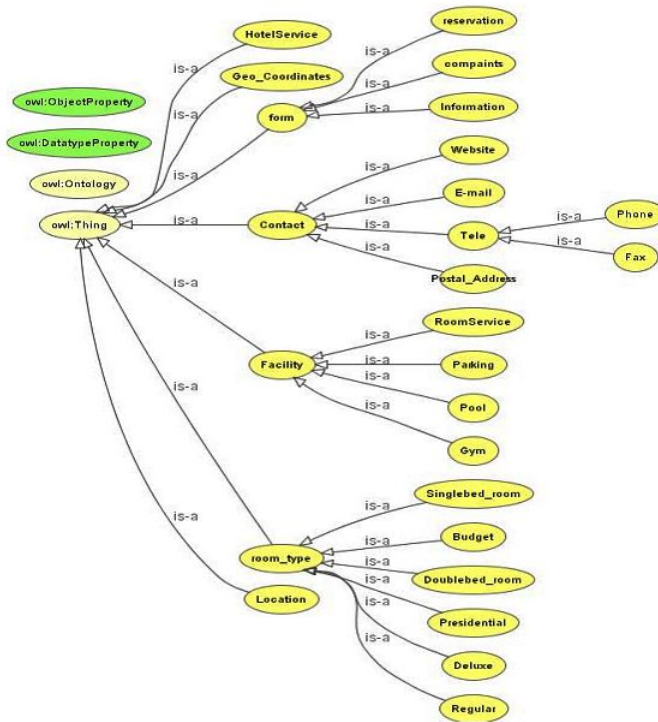


Fig. 6. Sample fragment of a Hotel-Service ontology.

4.5 Information Component

To provide information about a location on the map that is of user's interest, the use of Web Services is very desirable and our system uses Web browsing service. Also several Web services from XMethods¹¹ like zip code Web Service and Linear

¹¹ <http://www.xmethods.com/>


```

POST /lbsgisservice/Service1.asmx HTTP/1.1
Host: sice527.ddns.umkc.edu
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/getAddress"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAddress xmlns="http://tempuri.org/">
      <search>string</search>
      <location>string</location>
    </getAddress>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAddressResponse xmlns="http://tempuri.org/">
      <getAddressResult>string</getAddressResult>
    </getAddressResponse>
  </soap:Body>
</soap:Envelope>

```

Fig. 7. SOAP Request/Response Message.

distance service are used in the implementation. In our system the client receives and stores URLs that point towards Websites about the geographical locations that are shown on the client map. These are the URLs crawled by the server while crawling location information about the geographical locations. When the user clicks a location indicated on the map display, this component executes the Mobile Internet Explorer embedded in WinCE with the URL of the website for the geographical location.

5 Implementing Tour Guide System

Our system consists of client and server. We used iPAQ 5450(PocketPC) with GPS Receiver(Pretect CompactGPS) and wireless LAN card (802.11b) as the client and implemented the client application using Embedded Visual C++ 4.0. Web services wrapper over Google and information extraction application developed in MS Visual Studio .NET and query resolution tool made in Java as the server. The communication between the client and the server uses Wireless LAN(802.11 b). We implement SOAP-based Web Services based on .NET Framework and .NET Compact Framework. In the future work, we will be testing this system using CDMA network.

The client has two agents: Mobile GIS Agent and Tour Guide Agent. The Mobile GIS Agent handles the positioning component and the digital map service component to display the current location of the user received from GPS and the locations of attractions received from Tour Guide Agent. During this process, the communication component and information component are also involved to create/parse SOAP-based request/response message. The URLs are cached and the Websites will be processed when the user clicks on a geographical location on the digital map.

The server as mentioned previously is implemented including a Web service wrapper over the Google Web crawler and an Information extraction component that additionally includes a user-query resolution tool developed in Java. It uses Jena to query the Service Ontology and resolve the user request to closely matched establishments that may be found among the crawled service information. For instance, the Service Ontology is used to infer that the room is basically a property describing the concepts like hotel, inn, and hostels etc. which are all “places to stay”. The services extracted therefore contain services for inns, hotels etc. The user can further restrict his/her query by specifying that he/she needs a “cheap” room to stay. The concept of cheap room can be associated with hostels, motels etc. Thus the user’s query results would be restricted to hostels, motels etc. A sample output on a test client application is shown in Figure 8.

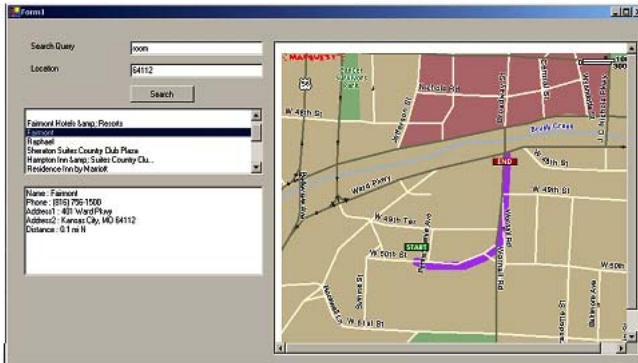


Fig. 8. An Example of Tour Guide Services.

6 Conclusion and Future Work

In this paper, we presented the efficient Mobile GIS and Web information discovery model for the Location Based Services in our Tour Guide System. We designed the architecture of the Tour Guide System. The advantages of this system include (1) the map services for Mobile GIS like small-sized spatial data, map scaling, multiple levels of detail, and computation of routing (2) location-based services to mobile users based on dynamically extracted information from the Web.

The combined capabilities of LBS and Web are still in their nascent stages. In order to fully access and use geospatial information with fully incorporated LBS services, it is important to extract semantically relevant geographical information from a wide range of geospatial data sources over the Web. There are significant research challenges that need to be addressed in semantic LBS services. Current Geographical Information sources are very limited and do not reflect the actual available services. There are the great challenges associated with the dynamic

features of the Web such as (1) location services need to be periodically updated (2) new services are continuously generated and (3) current services lack any standard representation format. Also from a system development perspective, since the mobile devices are limited in presenting the Web data, it is required to develop interactive and effective presentation techniques such as semantic caching, information filtering, user modeling.

Acknowledgement

This research was supported by the Program for the Training Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce, Industry and Energy of the Korean Government.

References

1. G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, M. Pinkerton, G. Marchionini, Cyberguide: A Mobile Context-Aware Tour. Technical Report GIT-96-06, Georgia Institute of Technology, (1997)
2. M. Califf and R. Mooney. Relation learning of pattern-match rules for information extraction. In Proc. 16th Nat. Conf. Artificial Intelligence, (1999)
3. N. Davis, K. Cheverst, K. Mitchell, A. Efrat, Using and Determining Location in a Context-Sensitive Tour Guide. *Cdcomputer*, Volume 34, Issue 8, IEEE, (2001) 35-41
4. J. W. Kim, S. S. Park, C. S. Kim, Y. Lee: The Efficient Web-based Mobile GIS Service System through Reduction of Digital Map. International Conference of Computational Science and Its Applications (ICCSA2004). Lecture Notes in Computer Science, Vol. 3043. Springer-Verlag, Berlin Heidelberg New York (2004) 410-417
5. N. Kushmerick, D. Weld, and R. Doorenbos.: Wrapper induction for information extraction. In Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI), (1997) 729-737
6. N. Mitra, SOAP Version 1.2 Part 0: Primer. W3C Recommendation, (2003)
7. C. Patel, K. Supekar, Y. Lee, OntoGenie: Extracting Ontology Instances from WWW. Workshop on Human Language Technology for the Semantic Web and Web Services. 2nd International Semantic Web Conference October 20th, (2003), pp. 127 - 130
8. T. Simcock, S. Hillenbrand, B. Thomas, Developing a Location Based Tourist Guide Application. Proceedings of the Australasian Information Security Workshop conference on ACSW Frontiers 2003, Australian Computer Society, (2003)
9. A. Stockus, A. Bouju, F. Bertrand and P. Boursier: Integrating GPS Data within Embedded Internet GIS. Proceedings of the 7th international symposium on Advances in geographic information systems (ACM GIS'99), ACM Press, (1999)
10. S. Takino: GIS on the fly to realize wireless GIS network by Java mobile phone. Web Information Systems Engineering, 2001. Proceedings of the Second International Conference, Volume 2, (2001)

A Progressive Reprocessing Transaction Model for Updating Spatial Data in Mobile Computing Environments

Donghyun Kim¹ and Bonghee Hong²

¹ Graduate School of Software, Dongseo University,
San 69-1, Churye-2 dong, Sasang gu, Busan, 617-716, Republic of Korea
pusrover@dongseo.ac.kr

² Department of Computer Engineering
Pusan National University
30 Jangjeon dong, Geumjeong gu, Busan, 609-735, Republic of Korea
bhhong@pusan.ac.kr

Abstract. Mobile transactions for updating spatial data are long-lived transactions that update local copies of the mobile platforms on disconnection. Since a mobile transaction is physically separated from its global transaction, the concurrent updates of mobile transactions should be merged into the global database after committing. Validation-based schemes, which are well-known to be appropriate for mobile transactions, have the overhead of aborting long duration transactions that conflict with some transactions. It is definitely unacceptable to cancel all the updates of a long-lived transaction due to conflicts with just a few objects. This paper introduces a novel reprocessing-transaction model that handles update conflicts between mobile transactions without aborting. Instead of aborting mobile transactions that conflict with committed transactions, the proposed model executes a new subtransaction called *a reprocessing transaction*, which reexecutes exactly the conflicted operations on conflicted objects with foreign conflicted objects. Foreign conflicted objects are part of the data committed by the other concurrent transactions and are related to the conflicted objects. We also propose *a progressive reprocessing scheme* to allow the non-conflicted objects of a mobile transaction to be incrementally exposed to other concurrent transactions in order to reduce the starvation of reprocessing transactions. Our reprocessing transaction model has the benefit of being able to serialize mobile transactions without aborting or waiting.

1 Introduction

New applications of mobile field systems [1], [2] recently have become more important for mobile users such as maintenance crews, inspectors, and surveyors. They use mobile platforms to collect up-to-date field data and keep as-built details on geographical map. Mobile clients in these applications first download the interested area from the server map database through a wireless network

and then perform field data updates on disconnection. They should connect to the server when it is required to process the commit operation [3]. Mobile transactions are usually long transactions that update the local copies stored in the mobile clients on disconnection.

Mobile transactions are initiated as local transactions to be executed at one or more mobile clients under the control of a global transaction. Since each mobile transaction is physically separated from the corresponding global transaction due to disconnection, the mobile transaction is allowed to be partially inconsistent with the global transaction before committing. One or more mobile transactions can be concurrently initiated for updating a given interested area. Furthermore, it is possible for more than one mobile transaction to update concurrently spatial objects in an overlapping region. To make updates of mobile transactions durable, their concurrent updates should be merged into the global database after committing. Unfortunately, we cannot detect any update conflicts before starting the merger phase. The problem with disconnection, which was uncovered in [4], is that when long-lived mobile transactions commit, aborting will likely occur.

Researchers on handling the conflicts of mobile transactions have experimented with three approaches. The first approach is to use the optimistic concurrency control scheme [5], [6], [7], [8]. This approach, generally, has been known to be appropriate for mobile transactions because mobile transactions can update the local copies of mobile clients without any communication with the server. The problem with the optimistic approach, however, is the aborting of long duration transactions. The second approach is "pessimistic" in that it relies on locking of the shared data. The critical problem with the pessimistic approach [9] is long waiting time caused by long duration transactions. The third is to use the cooperative working scheme [10] in wireless computing environments. This scheme, however, suffers from the overhead of merging histories between mobile clients and their station host whenever the mobile clients connect to the station host.

For mobile and long duration transactions, it is clear that they should not be aborted or delayed. This is the most important requirement for processing mobile transactions to interactively update spatial data. In this paper, the basic idea is to avoid aborting of conflicted mobile transactions performed for a long time and resolve the update conflicts between mobile transactions by reprocessing conflicted transactions under the modified validation-based scheme. When conflicts are detected between two mobile transactions, the **conflicted objects** of a newly committed transaction, first of all, are identified. In order to reduce the overhead to restart the conflicted transaction, we executed a new subtransaction called a **reprocessing transaction**, which reexecutes exactly the conflicted operations on the conflicted objects with their relevant foreign conflicted objects. **Foreign conflicted objects** are part of the write set of the already committed transactions and spatially related to the conflicted objects.

Our reprocessing scheme, however, can suffer from starvation. In this paper, we present a progressive reprocessing scheme to solve the starvation problem

of reprocessing transactions. Under this scheme, the reprocessing transaction exposes the non-conflicted objects of its write set to other concurrent mobile transactions.

In section 2, we discuss related works that handled update conflicts of mobile transactions. Section 3 defines the problems of validation-based protocols to control the interaction among mobile transactions. Section 4 describes a new extension of the existing validation protocols for handling indirect conflicts as well as direct conflicts between two write sets of mobile transactions. A reprocessing transaction scheme will be presented in section 5. Section 6 illustrates a procedure of recommitting for reprocessing conflicted transactions. Section 7 presents an implementation prototype of the reprocessing transaction model. Section 8 gives a summary and suggests further works.

2 Related Work

The first approach to the concurrency control of mobile transactions is to use the optimistic concurrency control scheme. In Mobisnap [5], when a mobile transaction commits, it is validated by testing the intended semantics described in pre-conditions and post-conditions. If one of the conditions is not satisfied, it is said that the transaction conflicts with others. For resolving conflicts, the transaction should be aborted. In [7], [8], they exploit the multi-version scheme for controlling concurrency of mobile transactions. A disconnected mobile transaction executes its operations on the locally copied snapshot. When the mobile transaction reconnects to its server and tries to commit, it performs a reconciliation process at the server. The reconciliation process consists of testing, conflict resolution and serialization. For resolving update conflicts, all the updates of the conflicted transaction are discarded.

In the two-tier transaction model [11], the first tier is a tentative transaction that works on the local data of a mobile node, and the second tier is a base transaction that is executed on those of a base node corresponding to a server. When a mobile node connects to the base node, all operations of a tentative transaction are transferred to the base node, and a base transaction reprocesses them. If the results of the base transaction do not satisfy some acceptance criteria, the transaction is aborted. If the base transaction fails, all the updates of the base transaction and the related tentative transaction are canceled. In [6], the history merge scheme is proposed using a base history and a tentative history. When a mobile client connects to a server, it transfers a tentative history to the server. The tentative history is tested for conflicts over the base history. The merger selects conflicted transactions from the tentative history and aborts them to resolve conflicts between two histories.

In order to resolve the update conflicts, all the schemes mentioned above abort conflicted transactions. It is obviously undesirable to abort conflicted transactions because mobile transactions updating the map data are long-lived transactions.

The second approach is to use the pessimistic concurrency scheme. In the mobile transaction model using locks [9], a mobile transaction should acquire a pre-read lock and a prewrite lock of an object. If a requested lock conflicts with the lock that is already set on the object, the mobile transaction is required to wait until it acquires all of the requested locks. However, when two mobile transactions update the shared spatial objects, this scheme suffers from a long waiting problem.

The third approach to resolving the update conflicts is to use the cooperative scheme. The CoAct model [10] presents an algorithm to merge the history of a mobile client with that of a server. To detect any conflicts, the merger searches for conflicted operations by scanning forward added histories. If the merger detects conflicts, it generates alternate merged histories in order to resolve the conflicts. It takes much time to detect conflicted operations by scanning forward and backward histories.

In order to avoid a long waiting time of conflicted mobile transactions, we use the optimistic concurrency control scheme. Instead of aborting the conflicted transactions, we present a reprocessing transaction model to resolve update conflicts by reprocessing conflicted objects. The benefit of our scheme is to guarantee serializability of mobile transactions without aborting or waiting under the optimistic assumption where the proportion of conflicted objects composing the entire written objects is considerably low.

3 Problem Definition

Mobile transactions for updating spatial objects differ from traditional transactions. First, mobile transactions are characterized by disconnection and reconnection of mobile clients. During disconnection, a mobile transaction can independently update its own local data without communicating with a server or other mobile clients. Second, updating of spatial data is usually interactive work on disconnection. Third, mobile transactions using wireless communication are long duration transactions on disconnection because of field-based updating of the copied map. In this paper, we focus on the issues of committing mobile transactions which have three characteristics: interactive update, long duration and disconnection.

Let us define the local copy of each transaction, T_i , to be the CopiedRegion(T_i) where $1 \leq i \leq n$. The Write and Read set of T_i can be defined as follows:

Definition 1: Suppose that $o_i.G$ is the spatial extent of an object o_i . **A Read Set**, denoted as $RS(T_i)$, is a set of objects that are contained within the CopiedRegion(T_i) and read by a read operation r of T_i . **A Write Set**, denoted as $WS(T_i)$, is a set of objects that are contained within the CopiedRegion(T_i) and written by a write operation w of T_i .

- $RS(T_i) = \{o_{ik} \mid \text{CopiedRegion}(T_i).G \cap o_{ik}.G \neq \emptyset \text{ and } r(T_i) \rightarrow o_{ik} \}$ where $1 \leq k \leq n$
- $WS(T_i) = \{o_{ik} \mid \text{CopiedRegion}(T_i).G \cap o_{ik}.G \neq \emptyset \text{ and } w(T_i) \rightarrow o_{ik} \}$ where $1 \leq k \leq n$

Let MBR be a minimum bounding rectangle including a set of objects o_{ik} , where $1 \leq k \leq n$, of T_i . For each transaction T_i , $\text{MBR}(\text{RS}(T_i))$ equals the $\text{CopiedRegion}(T_i)$.

The existing validation schemes check that T_i does not conflict with any other previously committed transactions in the Validation phase. Let us apply the existing validation scheme to mobile transactions for updating spatial data. There are two problems in processing mobile transactions by using the existing validation scheme. One is the cost of aborting conflicted transactions. If the validation test for a mobile transaction T_i fails, T_i has to discard all the objects of $\text{WS}(T_i)$. The cancel of long duration work is very costly and an undesirable operation for resolving the update conflicts of mobile transactions.

The other problem is that the intersection test, $\text{WS}(T_j) \cap \text{RS}(T_i) \neq \emptyset$, is too strict of a condition for testing the validation between two interleaving mobile transactions. Since spatial data are interactively updated, it is required to initially read and display all of the spatial objects within an interested area. Most objects of $\text{RS}(T_i)$ are accessed to display just them on each mobile client, but only a few $\text{RS}(T_i)$ are updated. Obviously, the intersection condition, $\text{WS}(T_j) \cap \text{RS}(T_i) \neq \emptyset$, may cause too many aborts.

Suppose that two transactions, T_1 and T_2 , update land parcels at two mobile clients. At the first mobile client, T_1 reads $p_1, p_2, p_{10}, p_{11}, p_{12}, p_{15}, p_{16}, p_{17}$ and p_{19} to display just them for performing an interactive update as shown in Fig. 1. Fig. 2 shows a scenario of concurrently updating the two different write sets of T_1 and T_2 .

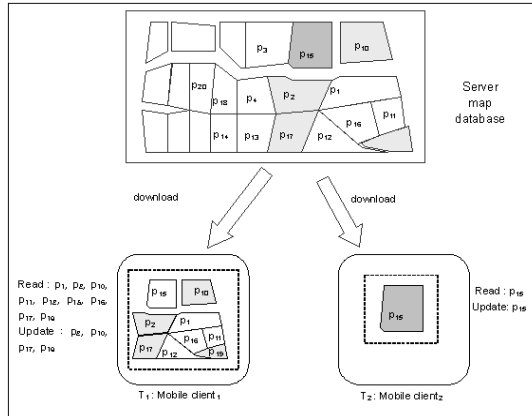


Fig. 1. Example of two mobile transactions, T_1 and T_2 .

Suppose that the transaction order is set to be $\text{start}(T_1) < \text{start}(T_2) < \text{commit}(T_2) < \text{commit}(T_1)$. T_1 has a long read phase. When T_1 is validated, T_2 has already completed its validation phase before T_1 commits. In the case of T_1 , which commits its long update work, its validation test fails because $\text{RS}(T_1) \cap \text{WS}(T_2) = p_{15}$, and T_2 commits before T_1 . Finally, T_1 is aborted and has

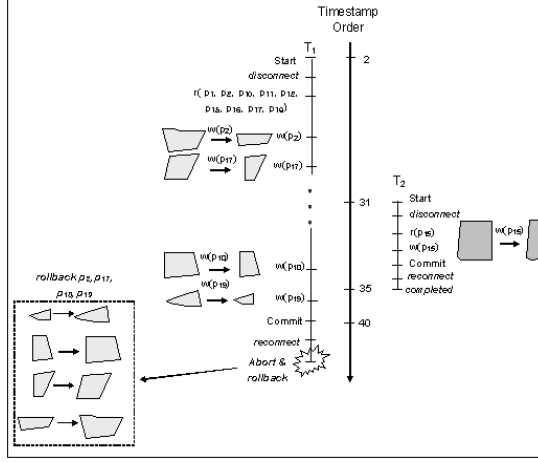


Fig. 2. Example of updating two write sets concurrently.

to cancel its long duration updates, p_2 , p_{10} , p_{17} and p_{19} . However, T_1 should not be aborted because there are no actual update conflicts between T_1 and T_2 , where the two write sets of T_1 and T_2 actually differ from each other. The read of p_{15} does not cause any critical concurrency problem as shown in Fig. 1 and Fig. 2.

One last consideration is a possibility of getting spatial inconsistencies between two different spatial objects. In a spatial database, an update operation of an object o_n can affect the spatial consistency over the other object o_m where the two objects have non-disjoint spatial relationships [12]. The necessary condition of being spatially inconsistent is defined as follows: $o_n \neq o_m$ and $o_n.G \cap o_m.G \neq \emptyset$ for $o_m \in WS(T_i)$ and $o_n \in WS(T_j)$. When two distinct but spatially non-disjoint objects are updated independently, the result of merging two updated objects may become inconsistent. This indirect conflict, so far, has not been handled under the existing validation schemes.

To detect spatial inconsistency, we define the concept of *indirect conflict* of mobile transactions for updating spatial data as follows.

Definition 2: Given two write sets $WS(T_i)$ and $WS(T_j)$ where $start(T_i) < commit(T_j) < commit(T_i)$, we can say that o_n can *indirectly conflict* with o_m iff $o_n \neq o_m$ and $o_n.G \cap o_m.G \neq \emptyset$ for $\exists o_n \in WS(T_i)$ and $\exists o_m \in WS(T_j)$.

4 Validation Conditions for Detecting Update Conflicts

This section describes some extended validation conditions for detecting indirect conflicts as well as direct conflicts between the write sets of two mobile transactions.

4.1 Interleaved and Overlapped Transactions

When a mobile transaction T_i commits, it should perform the validation test against the write sets of early committed transactions T_j where $\text{start}(T_i) < \text{commit}(T_j) < \text{commit}(T_i)$. We call T_j an interleaved transaction of T_i and define it as follows:

Definition 3: Given two mobile transactions, T_i and T_j , T_j is an *interleaved transaction* of T_i iff $\text{start}(T_i) < \text{commit}(T_j) < \text{commit}(T_i)$.

For the validation test of T_i , it needs to find out all the interleaved transactions of T_i . When T_i is validated, all the write sets of transactions that were completed before T_i must be examined. We denote the list of interleaved transactions of T_i as $\text{validation_list}(T_i)$.

If the copied region of T_i does not intersect with that of T_j where T_i and T_j are interleaved transactions, we don't need to perform the validation test between T_i and T_j . Let us consider the minimal bounding region for covering the read set of T_i as the copied region of T_i . Using the minimal bounding region of the local copy of each mobile transaction, we can define *overlapped transactions* as follows:

Definition 4: Given two interleaved transactions, T_i and T_j , T_i and T_j are *overlapped transactions* iff $\text{CopiedRegion}(T_i).G \cap \text{CopiedRegion}(T_j).G \neq \emptyset$.

If T_i and T_j are interleaved and overlapped, they may conflict with each other.

Definition 5: Given two mobile transactions, T_i and T_j , T_i might *potentially conflict* with T_j iff T_i is both interleaved and overlapped with T_j .

To improve the concurrency of mobile transactions, it is very important to reduce the number of potentially conflicted transactions. By using the concept of overlapped transactions, $\text{validation_list}(T_i)$ can be further refined as follows: $\text{validation_list}(T_i) = \{T_j \mid \text{start}(T_i) < \text{commit}(T_j) < \text{commit}(T_i) \text{ and } \text{CopiedRegion}(T_i).G \cap \text{CopiedRegion}(T_j).G \neq \emptyset\}$ where $i \neq j$.

4.2 Detecting Conflicted Objects

The intersection test between the write sets and the read set of two mobile transactions, T_i and T_j , differs from that of short transactions. In order to just display the spatial objects within a given region, the read set of a transaction T_i , $\text{RS}(T_i)$, should be read to a mobile client. The conflicts between the write set of T_j and the read set of T_i are not critical as already shown in Fig. 1 and Fig. 2.

To avoid aborting of conflicted transactions, it is important to detect conflicted objects between two write sets. The first write set is said to be conflicted with the second write set if the same object is read and written by two transactions, or written by them. Besides, mobile transactions to update spatial data have an idiosyncrasy of spatial consistency between spatial objects. A validation test should be done for satisfying spatial predicates to detect conflicts between

spatial objects, although they are spatially disjointed. The existing validation condition therefore should be extended for detecting indirect conflicts as follows.

Definition 6: For any mobile transaction T_i , the *extended validation conditions* of T_i include not only **Condition1** but also **Condition2**:

- **Condition1(direct conflict condition):** $\text{start}(T_i) < \text{commit}(T_j) < \text{commit}(T_i)$ and $\text{WS}(T_i) \cap \text{WS}(T_j) \neq \emptyset$
- **Condition2(indirect conflict condition):** For $\exists o_m \in \text{WS}(T_i)$ and $\exists o_n \in \text{WS}(T_j)$, $\text{start}(T_i) < \text{commit}(T_j) < \text{commit}(T_i)$ and $\text{WS}(T_i) \cap \text{WS}(T_j) = \emptyset$ and $o_m.G \cap o_n.G \neq \emptyset$

One of the above two conditions must hold for the extended validation conditions to fail. For detecting indirect conflicts, T_i needs to check the spatial intersection between two updated objects in the validation phase. The spatial intersection can be tested by using both the old extent and the new extent of a spatial object. Suppose two objects, o_m and o_n , are updated where $o_m \in \text{WS}(T_i)$ and $o_n \in \text{WS}(T_j)$. Let $o_m.G_{\text{new}}$ and $o_m.G_{\text{old}}$ denote the new extent and the old extent of o_m respectively. If $o_m.G_{\text{old}} \cap o_n.G_{\text{old}} \neq \emptyset$ or $o_m.G_{\text{new}} \cap o_n.G_{\text{new}} \neq \emptyset$ or $o_m.G_{\text{old}} \cap o_n.G_{\text{new}} \neq \emptyset$, then $o_m.G$ is said to be *spatially intersected* with $o_n.G$. By means of the extended validation condition, a set of conflicted objects of T_i is defined as follows.

Definition 7: Given $\text{WS}(T_i)$ and $\text{WS}(T_j)$ where T_j is in the `validation_list` of T_i , the *conflicted objects* of T_i are the objects of T_i that are common to both $\text{WS}(T_i)$ and $\text{WS}(T_j)$ or the indirectly conflicted objects of $\text{WS}(T_i)$ whose the extent intersects with the extent of the objects of $\text{WS}(T_j)$.

- For $\exists o_m \in \text{WS}(T_i)$ and $\exists o_n \in \text{WS}(T_j)$, $\text{conflicted_objects}(T_i) = \{o_m \mid o_m.\text{oid} = o_n.\text{oid} \text{ or } o_m.G \cap o_n.G \neq \emptyset\}$

5 Reprocessing Conflicted Objects

We describe a new reprocessing scheme for handling the conflicted objects of which basic idea is to use the concept of foreign conflicted objects in order to resolve the update conflict of mobile transactions.

5.1 Foreign Conflicted Objects

The basic idea to reprocess conflicted transactions is to separate the write set of a committing transaction into two parts: one is non-conflicted objects and the other is conflicted objects. The non-conflicted objects are consistent with the global database state because they do not conflict with any objects of other concurrent transactions. Thus, it is possible to expose them to newly arriving transactions. The conflicted objects, meanwhile, can not be exposed to the others because they potentially conflict with other updates. Instead of undoing the conflicted transaction, the conflicted objects should be arranged to be reprocessed by a subtransaction of the conflicted transaction.

Conflicts of interactive updates on spatial data should be resolved by the user's decision. However, the reprocessing step to resolve the update conflicts needs to know which objects of the committed transactions cause conflicts. Suppose T_i is checked against the previously committed transaction T_j . When an object o_{i1} of T_i conflicts with o_{j1} of T_j , o_{i1} is named as *conflicted object* of T_i , and o_{j1} is named as *foreign conflicted object* of T_i . The foreign conflicted objects of T_i are defined as follows.

Definition 8: Given $\text{conflicted_objects}(T_i)$ and $\text{WS}(T_j)$ where $\text{start}(T_i) < \text{commit}(T_j) < \text{commit}(T_i)$, for $\exists o_{im} \in \text{conflicted_objects}(T_i)$ and $\exists o_{jk} \in \text{WS}(T_j)$, ***foreign_conflicted_objects(T_i)*** = $\{ o_{jk} \mid o_{im}.G \cap o_{jk}.G \neq \emptyset \}$

5.2 A Reprocessing Transaction

We refer to the reprocessing step for resolving the conflicted objects of T_i as the reprocessing transaction of T_i , namely RT_i . RT_i is a subtransaction which is initiated by the conflicted transaction T_i and reupdates only the conflicted objects of T_i . First of all, a reprocessing transaction needs to bring the foreign conflicted objects to the mobile client for interactively correcting the conflicted objects. Unlike the conflicted transaction T_i , a reprocessing transaction RT_i has two kinds of the read set: the conflicted objects of T_i and the foreign conflicted objects of T_i .

Since a reprocessing transaction is dedicated to resolve the update conflicts, RT_i should reupdate exactly the conflicted objects of T_i by user's decision. The write set of RT_i , $\text{WS}(\text{RT}_i)$, is $\text{conflicted_objects}(T_i)$. The non-conflicted objects of T_i , in the meantime, will be immediately exposed to other transactions before starting RT_i . This is an important consideration for progressively reprocessing update conflicts, which will be described in detail in the next section.

Let the k^{th} reprocessing transaction of T_i denote RT_{ik} . When RT_{ik} commits, the write set that are reupdated by RT_{ik} should be checked against newly arriving transactions that have been committed during RT_{ik} 's execution. If any conflicts occur once again, the next reprocessing transaction $\text{RT}_{i(k+1)}$ starts. For example, a new transaction T_k commits before RT_{i1} commits, as shown in Fig. 3. If RT_{i1} conflicts with T_k , RT_{i1} should be reprocessed for handling the conflict with T_k . At this time, the second reprocessing transaction RT_{i2} is a subtransaction of the first reprocessing transaction RT_{i1} . The validation test is performed for the write set of RT_{i1} with the write set of T_k . The conflicted objects of RT_{i1} are a subset of the original conflicted objects of T_i and selected by testing the intersection between the conflicted objects of T_i and the write set of T_k . The foreign conflicted objects of RT_{i1} are part of the write set of T_k that indirectly conflicts with RT_{i1} . The read set of RT_{i2} is the union of the conflicted objects of RT_{i1} and the foreign conflicted objects of RT_{i1} .

A reprocessing transaction may be executed repeatedly. If the k^{th} reprocessing transaction continuously conflicts with newly arriving transactions, there is a possibility of starvation due to a sequence of conflicting transactions that cause repeated restarts of reprocessing transactions. Fortunately, our reprocess-

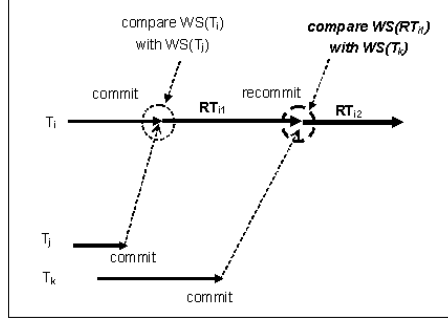


Fig. 3. Reprocessing of the reprocessing transaction RT_{i1} .

ing transaction model can progressively expose non-conflicted objects to newly committed transactions, which will be described in detail in the next section.

6 Recommit Processing

In this section, we describe a procedure for processing recommit in order to reduce starvation of reprocessing transactions.

6.1 Progressive Reprocessing

During execution of the k^{th} reprocessing transaction RT_{ik} , a new transaction T_j or the other reprocessing transaction RT_{jl} may commit. For example, when a reprocessing transaction RT_{11} completes the reupdating of the conflicted objects of T_1 (see Fig. 4), the validation condition for RT_{11} should be checked against T_3 and T_4 . If the write set of RT_{11} conflicts with that of T_4 , what should be done for recommitting RT_{11} ?

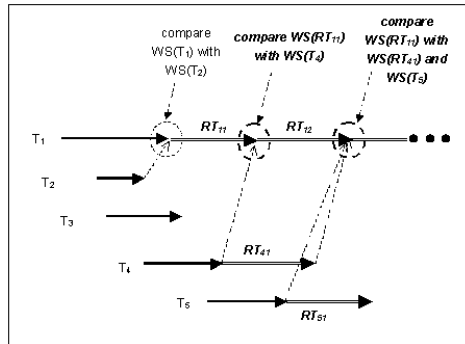


Fig. 4. Example of repeated reprocessing.

We consider how to reprocess the reprocessing transaction that conflicts again with newly arriving transactions. The first approach to deal with this problem is a repeated reprocessing scheme. A reprocessing transaction RT_{ik} compares the original conflicted objects of T_i with the write set of a newly committed transaction T_j when RT_{ik} recommits. If RT_{ik} conflicts with T_j , the next reprocessing transaction is created for reprocessing the conflict of RT_{ik} with T_j . This approach, however, suffers from starvation. Under the repeated reprocessing scheme, the conflicted transaction T_i continuously may be reprocessed due to recently committed transactions.

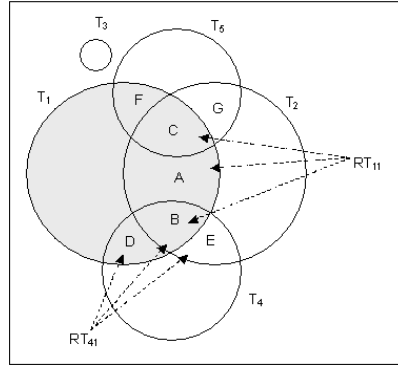


Fig. 5. Intersection diagram of interleaved, overlapped transactions.

For example, as shown in Fig. 4, the conflicted objects between T_1 and T_2 are checked against T_4 , when RT_{11} recommits. The dotted arrow in Fig. 4 denotes update conflicts. A diagram in Fig. 5 shows the intersection diagram among the write sets of T_1 , T_2 , T_3 , T_4 , T_5 , RT_{11} and RT_{41} . The write set of RT_{11} is $WS(T_1) \cap WS(T_2)$ which is $\{A, B, C\}$ in Fig. 5. RT_{11} cannot be completed because RT_{11} conflicts with T_4 . At this time, RT_{11} needs to initiate the next reprocessing transaction, RT_{12} , for resolving its conflicts with T_4 . When T_4 commits, the write set of T_4 intersects with that of T_1 and T_2 . This leads to reprocessing of T_4 against T_1 and T_2 . When RT_{11} commits, the validation test is performed against T_4 . Because the intersection among T_1 , T_2 , and T_4 is $\{B\}$ as shown in Fig. 5, the intersection between RT_{11} and T_4 is also $\{B\}$. In this case, it is required to reprocess the original conflicted objects of T_1 with T_4 . In the next place, a new transaction T_5 commits during the execution of RT_{12} . The original conflicted object of T_1 should be reprocessed once again because the intersection between $WS(RT_{12})$ and $WS(T_5)$ is $\{C\}$.

In order to avoid the starvation problem, we propose a progressive reprocessing scheme that allows other transactions to access part of the write set of the reprocessing transaction after each reprocessing step. If the k^{th} reprocessing transaction RT_{ik} conflicts with a newly committed transaction T_j , the $k+1^{th}$ reprocessing transaction $RT_{i(k+1)}$ will be generated to resolve the conflicted ob-

jects of RT_{ik} with T_j . The conflicted objects of RT_{ik} are part of the conflicted objects of the $k-1^{th}$ reprocessing transaction. The $RT_{i(k+1)}$ reupdates only the conflicted objects of RT_{ik} with T_j . Before starting $RT_{i(k+1)}$, RT_{ik} commits partially non-conflicted objects of its write set and exposes them to other transactions. The progressive reprocessing scheme allows later transactions to access the non-conflicted objects of the write set, even if the reprocessing transaction is not completely finished.

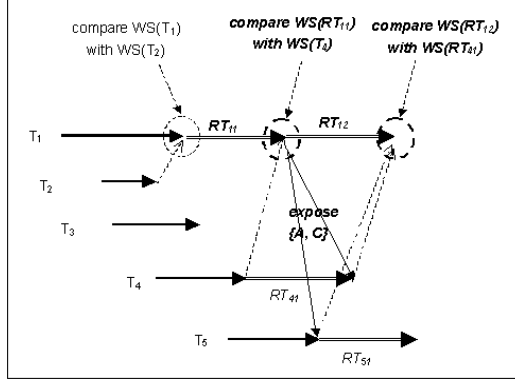


Fig. 6. Example of progressive reprocessing.

For example, when RT_{11} completes its reprocessing work in Fig. 6, the write set of RT_{11} is validated against a newly committed transaction T_4 . As shown in Fig. 5, part of the write set of R_{11} , $\{B\}$, conflicts again with T_4 . The other part, $\{A, C\}$, does not conflict with T_4 . RT_{12} reprocesses only $\{B\}$, which is also part of the conflicted objects of T_1 . When RT_{11} completes its recommit, RT_{11} exposes $\{A, C\}$ to RT_{41} and T_5 . As the reprocessing transaction reprocesses the conflicted objects progressively, the conflicted objects for reprocessing diminish gradually. All of the conflicted objects of T_i can be resolved in the result.

6.2 Recommit Processing Against Ongoing Transactions

When the k^{th} reprocessing transaction RT_{ik} recommits, the validation test should be done against ongoing transactions. We consider two kinds of recommits as shown in Fig. 7. As described in section 6.1, when RT_{11} recommits, T_4 has been in the state of reprocessing by means of the *primary reprocessing* of T_4 , RT_{41} . The write set of RT_{11} intersects with that of RT_{41} , according to Fig. 5. The validation of RT_{11} requires another reprocessing transaction to correct part of the conflicted objects of T_1 (i.e. $WS(RT_{11}) \cap WS(T_4) = \{B\}$). When RT_{41} recommits, RT_{12} is performing the secondary reprocessing of $\{B\}$. How do the ongoing transactions affect the recommit operation of a reprocessing transaction?

The first case of recommitting is the validation test against the presumed committed transactions whose write set may conflict with the write set of the

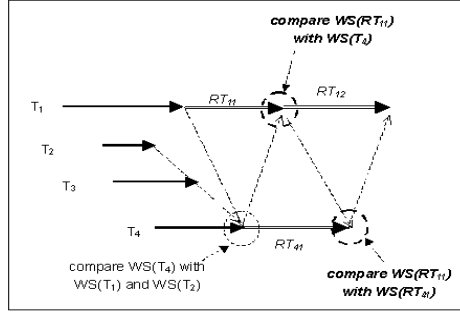


Fig. 7. Two cases of recommitting.

primary reprocessing transaction. Even if the reprocessing transaction of T_j , RT_{j1} , is reupdating conflicted objects of T_j , the k^{th} reprocessing transaction of T_i , RT_{ik} , needs to compare its write set with the write set of T_j . The reconflict of the write set of the primary reprocessing transaction is forced to be reprocessed due to later committed transactions.

The second case of recommitting is the validation test against the m^{th} reprocessing transaction of T_j , RT_{jm} where $m > 1$. If $WS(RT_{jm}) \cap \text{conflicted_objects}(RT_{i(k-1)}) \neq \emptyset$ and $\text{CopiedRegion}(RT_{jm}).G \cap \text{CopiedRegion}(RT_{i(k-1)}) \neq \emptyset$, the recommitting transaction RT_{ik} should determine acceptance of the results of the other reprocessing transaction RT_{jm} . For example, when RT_{41} recommitting, the validation test is performed against RT_{11} . RT_{12} of RT_{11} was invoked because of the conflicts with T_4 . RT_{41} has two conflicting parts of the intersection between T_1 and T_4 . One is $\{D\}$ which is not related to RT_{11} . The other is $\{B\}$ which is $WS(RT_{11}) \cap \text{conflicted_objects}(T_4)$. The reconflict of $\{B\}$ can be finally resolved by merging the write set of RT_{41} with the write set of RT_{11} at the time of recommitting RT_{41} . If RT_{41} accepts the results of RT_{11} 's reupdating, RT_{12} will be a useless reprocessing transaction. If RT_{41} does not accept RT_{11} 's results, the update conflicts should be harmonized by RT_{12} or its next reprocessing transaction. This is a remaining overhead of our reprocessing approach to mobile transaction processing.

7 Implementation

In this section, we present an implementation prototype of a mobile transaction manager to realize our reprocessing transaction model.

7.1 A System Architecture of a Mobile Transaction Manager

Fig. 8 shows an overview of the prototype system of a mobile transaction manager. The implemented mobile transaction server is built on the spatial data server, called 'CyberMap'[18], running on the Linux server. The server exploits the IMT2000x1 wireless network in order to talk with its clients for processing mobile transactions.

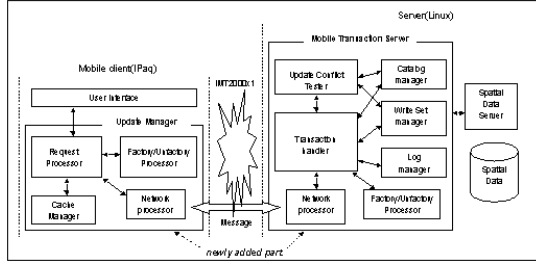


Fig. 8. A system architecture of a mobile transaction manager.

7.2 Example of the Reprocessing Transactions

A mobile transaction, first of all, should display all the spatial objects within a given region in order to interactively update some spatial objects. Fig. 9 shows the initial display of spatial objects in each mobile transaction at the client side.

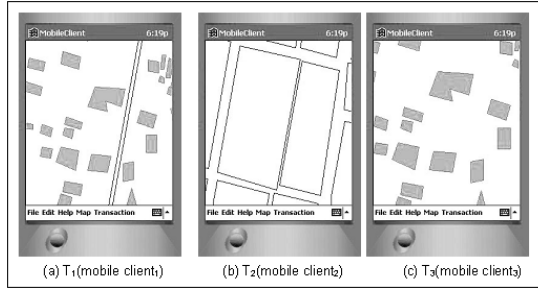


Fig. 9. The original display of spatial objects in each mobile transaction.

Suppose that the transaction order is set to be $\text{commit}(T_2) < \text{commit}(T_1) < \text{commit}(T_3) < \text{commit}(RT_{11})$. T_1 updates two objects of a house (see Fig. 10(a)), and T_2 updates one object of a road (see Fig. 10(b)). They are disparate objects. When T_1 performs the validation test against the write set of T_2 , the test fails because two objects of T_1 have an intersection relationship with the object of T_2 . T_1 generates $\text{conflicted_objects}(T_1)$ and $\text{foreign_conflicted_objects}(T_1)$ and starts its reprocessing transaction RT_{11} . As shown in Fig. 10(c), RT_{11} highlights $\text{conflicted_objects}(T_1)$ and displays $\text{foreign_conflicted_objects}(T_1)$. The dotted object is $\text{foreign_conflicted_objects}(T_1)$.

RT_{11} updates $\text{conflicted_objects}(T_1)$ in order to resolve conflicts(see Fig. 11(a)) and recommits. In Fig. 11(b), T_3 updates one object of the house and commits successfully during RT_{11} 's execution. Since the write set of RT_{11} conflicts again with the write set of T_3 , RT_{11} starts RT_{12} . RT_{12} displays $\text{conflicted_objects}(RT_{11})$ and $\text{foreign_conflicted_objects}(RT_{11})$ in the mobile client as shown in Fig. 11(c).

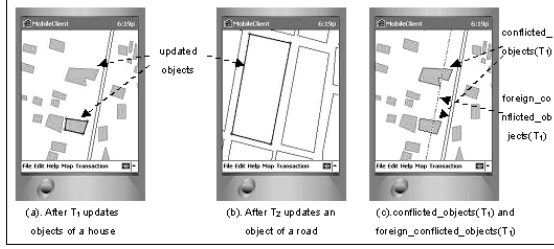


Fig. 10. Conflicts between T_1 and T_2 .



Fig. 11. RT_{11} recommit.

The non-conflicted object of RT_{11} is exposed to the other transactions in Fig. 12(a) and RT_{12} reexecutes only one object, which is $\text{conflicted_objects}(RT_{11})$. As shown in Fig. 12(b), RT_{12} recommit successfully, and T_1 is finally committed. Fig. 12(c) shows a final state where all conflicts are resolved properly.

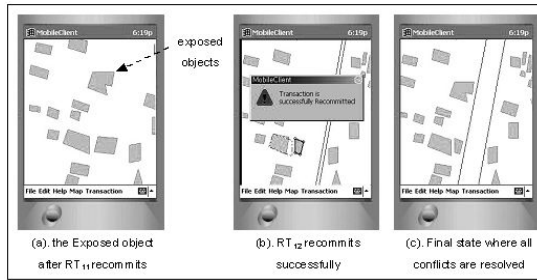


Fig. 12. RT_{12} recommit.

8 Conclusions

Validation-based protocols come naturally to process mobile transactions, which are independently carrying out their own interactive and long duration updates during disconnection. We tried to use the existing optimistic approach based

on the validation protocols to eliminate the overhead of locking. Under the validation-based protocols, a new difficulty arose: should we lose a long-lived transaction's work when some validation conditions fail? The motivation of this work is to avoid aborting conflicted transactions and reprocess only conflicted objects.

In this paper, we have proposed a reprocessing transaction model to resolve the update conflicts without waiting or aborting when the validation test fails. The reprocessing of update conflicts can be easily achieved by reexecuting the conflicted objects using the foreign conflicted objects which can be computed from the write sets of previously committed transactions. A new difficulty in the reprocessing transaction model is the repeated occurrence of update conflicts with newly arriving transactions when a reprocessing transaction recommits. This problem can be handled by a progressive reprocessing scheme in order to reduce the starvation of reprocessing transactions. The progressive reprocessing scheme allows part of the write set of the reprocessing transaction to be exposed to other transactions during reprocessing. We have also shown an implementation overview of the prototype system for realizing a reprocessing transaction model.

Our reprocessing transaction model is absolutely superior to the aborting method of the validation-based protocols, if the ratio of intersection between two write sets for all objects of the write sets is relatively low. Further studies are needed to consider the case of too much intersection overlap between two conflicted transactions and to reduce useless reprocessing transactions.

References

1. Dong Hyun Kim, Bong Hee Hong, Byunggu Yu, Eun Suk Hong: Validation-Based Reprocessing Scheme for Updating Spatial Data in Mobile Computing Environments, *Int. Conf. on advanced information networking and application*, pp.211-214, 2003.
2. Field Solutions: Technical White Paper, Tadpole Technology, Inc.
3. James J. Kistler, M. Satyanarayanan: Disconnected Operation in the Coda File System, *ACM Transaction on Computer System*, Vol. 10, No. 1, pp. 3-25, 1992.
4. Margaret H. Dunham, Abdelsalam Helal, and Santosh Balakrishnan: A Mobile Transaction Model that Captures Both the Data and Movement Behavior, *Mobile Networks and Applications*, Vol. 2, Issue 2, pp.149-162, 1997.
5. Nuno Pregoica, Carlos Baquero: Mobile Transaction Management in Mobisnap, *Advances in Databases and Information Systems*, pp.379-386, 2000
6. Peng Liu, Paul Ammann, Sushil Jajodia: Incorporating Transaction Semantics to Reduce Reprocessing Overhead in Replicated Mobile Data Applications, *Int. Conf. on Distributed Computing Systems*, pp.1-10, 1999.
7. Shirish Hemant Phatak, B. R. Badrinath: Multiversion Reconciliation for Mobile Databases, *Int. Conf. on Data Engineering*, pp.582-289, 1999.
8. Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, Patrick O'Neil: A Critique of ANSI SQL Isolation Levels, *Proc. of the ACM SIGMOD*, pp.1-10, 1995.
9. Sanjay Kumar Madria, Bharat Bhargava: A Transaction Model for Mobile Computing, *Int. Database Engineering and Application Symposium*, pp.92-102, 1998.

10. Justus Klingemann, Thomas Tesch, Jurgen Wasch: Enabling Cooperation among Disconnected Mobile Users, *Int. Conf. on Cooperative Information Systems*, pp.36-45, 1997.
11. Jim Gray, Pat Helland, Patrick O'Neil, Dennis Shasha: The Dangers of Replication and a Solution, *Proc. of the ACM SIGMOD*, pp.173-182, 1996.
12. Sylvie Servigne, Thierry Ubeda, Alain Puricelli, Robert Laurini: A Methodology for Spatial Consistency Improvement of Geographic Databases, *GeoInformatica*, Vol. 4, pp.7-34, 2000.
13. Maher Suleiman, Michele Cart, Jean Ferrie: Concurrent Operations in a Distributed and Mobile Collaborative Environment, *Int. Conf. on Data Engineering*, pp.36-45, 1998.
14. Henry F. Korth, Greg Speegle: Formal Aspects of Concurrency Control in Long duration Transaction Systems Using the NT/PV Model, *ACM Transaction on Database Systems*, Vol. 19, No. 3, pp.492-535, 1994.
15. Abraham Silberschatz, Henry F. Korth, S. Sudarshan: *Database System Concepts*, McGraw-Hill Companies, 1996.
16. Jim Gray, Andreas Reuter: *Transaction Processing: Concepts and Technique*, Morgan Kaufmann Publishers, 1997.
17. Gerhard Groger, Lutz Plumer: Provably Correct and Complete Transaction Rules for GIS, *ACM Int. Symposium on Advances in Geographic Information Systems*, pp.40-43, 1997.
18. CyberMapWorld: Web GIS Services, www.gilmap.com

Mediation for Online Geoservices

Omar Boucelma and François-Marie Colonna

LSIS UMR CNRS 6168 and Université Paul Cézanne Aix-Marseille III
Avenue Escadrille Normandie-Niemen
F-13397 Marseille Cedex 20
{Omar.Boucelma,Francois-Marie.Colonna}@lsis.org

Abstract. Interoperating Geographic Information Systems (GIS) poses several challenges. First, despite OpenGIS Consortium recommendations, GML is an emerging standard. Second, each GIS provides its own proprietary format as well as its specific query language; while geographic resources are designed for a variety of different purposes. Finally, orthogonal directions in the design of geographic resources may affect the semantics of the data they contain and impair their integration.

With the proliferation of GIS data and resources over the Internet, there is an increasing demand for robust geospatial information services that allow federation/interoperation of massive repositories of heterogeneous spatial data and metadata.

The purpose of this paper is to show how *mediation* – a data integration technique – can help in building such a Web-based required geospatial service. This technique has been fully implemented in the context of a geographic mediation/wrapper system that provides an integrated view of the data together with a spatial query language. As a proof of concept, we deployed the service in building a prototype for an interoperability application involving several catalogues of satellite images.

1 Introduction

With the widespread use of commercial or open sources GIS, amount of spatial data has grown exponentially. The creation of internet made these ressources available on the web, but unable to communicate with each other. Interoperability of these sources is next GIS challenge to step, as geographic applications can be accessed from a web browser, or even a mobile device such as a cellular phone. The data integration technique called *mediation* can help in building such a Web-based required geospatial service. In this paper, we present our solution for building a web based mediator, and show an example involving catalogues of satellite images.

Typical mediation approaches are data-driven and do not address the problem of integration of query capabilities. But the exploitation of available query capabilities is critical to a geographic mediation system. Geographic languages provided by GIS usually express spatial selections, metric or topological queries, allocations, etc., in addition to standard data manipulation such as performed by SQL[17], and are usually implemented as ad-hoc functions with respect to a

specific data representation and indices. These query capabilities may be available partially or totally at some of the integrated data sources. Similar operators may not be semantically equivalent at two different sources.

Our main contributions are:

- VirGIS provides a transparent non-materialized integrated view of geographic data.
- VirGIS uses GML [12] as an internal format to represent, manipulate, and exchange geographic information.
- we purposely developed GQuery, a XML geographic query language based on XQuery.
- VirGIS exploits querying capabilities available at integrated GIS, and extends them with local capabilities when the needed capability is missing at the source but available locally [6].
- VirGIS accesses integrated GIS features and query capabilities via Web Feature Server (WFS) [14]. A WFS query consists of a SQL-like query language called CQL [15], embedded in a XML program.
- Geographic queries are expressed either with GQuery, or via a WFS query expression.
- VirGIS complies with recent standard specifications of the Open GIS Consortium [15]. In addition these standards are being adopted by major GIS vendors [1, 2].

The paper is organized as follows: Section 2 discusses our approach to query GML data. Section 3 gives an overview of the mediation system, while Section 4 describes a concrete (and real) application scenario. Finally, we conclude in Section 5.

2 Querying GML Data

2.1 GML Data Model

In this section, we give an overview of GML key concepts as described by OGC Abstract Specification [15]. The basic concept is a *Feature*, i.e., an (object) abstraction of the real world phenomena, with spatial and non-spatial attributes. Figure 1 shows a town split into four districts.

Figure 2 illustrates the UML schema of a town, according to the OpenGIS abstract model. Town and parcels classes inherit from Feature, and parcel has a Geometric property.

A simple encoding of these features with GML 2.0 (Figure 3) only shows containment relationships between town and districts, as parcels tags are nested in town tag. Not all information represented in Figure 1 is available in GML. For example, adjacency relationships between parcels need to be first defined in an XML schema, and then encoded in the GML document. When working with complex examples of thousands of features, expressing all relationships between them is prohibitive because it increases GML document complexity and

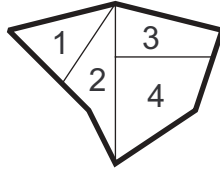


Fig. 1. Town Districts.

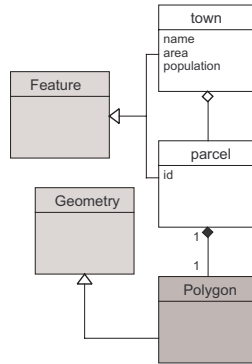


Fig. 2. UML Modeling of a Town Districts.

size. The new release GML 3.0 has the ability to store topological relationships between features with new tags (see the detailed example in [13], page 468). The inconvenient of this encoding is also the size of the document. A smaller one will be transferred faster on the network. What is needed is a query language for GML with spatial operators that captures the spatial semantics from the smallest GML encoding.

```
<town name='NY' area='500' population='10000'>
  <parcel id='1'><polygon>...</polygon></parcel>
  ....
  <parcel id='4'><polygon>...</polygon></parcel>
</town>
```

Fig. 3. District GML 2.0 Encoding.

2.2 Related Work on Querying GML

Querying spatial data is a well known problem studied in the context of SQL and relational DBMS and resulted in languages such as Spatial SQL[8] or GeoSQL [9] or Oracle Spatial, all of them being SQL extensions. The choice of extending an existing language, instead of creating a new one is motivated by the fact that spatial databases contain both spatial and non-spatial data. Querying GML data could be performed in a similar way, i.e., the query language should be developed on the basis of the languages developed for XML.

The OpenGIS consortium, with its Web Feature Service requests (WFS) [15], provides a way to specify data manipulation operations on geographic features posted to a Web Feature Server using HTTP requests (Figure 4). The Web Feature Server layer is in charge of manipulating the data sources that may contain geographic features.

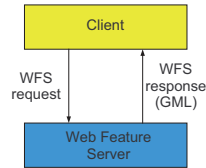


Fig. 4. Web Feature Service.

Web Feature Service interfaces allow geographic data extraction that is returned to the user as a GML document. When accessing a source, one may execute spatial operators provided by the underlying GIS. There are mainly two kind of requests (GET and POST). The examples below query all instances of type “TOWN” in the spatial database. The first method uses keyword-value pairs (KVP) to encode the various parameters of a request as shown in Figure 5.

```
http://www.someserver.com/wfs.cgi?service=WFS &version=1.0.0
&request=GetFeature &typename=TOWN
```

Fig. 5. WFS query by http GET.

```
<?xmlversion="1.0"?> <GetFeature version="1.0.0" service="WFS">
  <Query typeName="TOWN"/>
</GetFeature>
```

Fig. 6. WFS query by http POST.

The second method uses a specific XML encoding as a query language as shown in Figure 6. In both cases, HTTP URLs or XML encoded WFS requests can express selections of data. Their expressive power is not equivalent to SQL or XQuery, and it is not easy to write and read these queries. WFS is also not able to join data from two distinct GIS. Some spatial operators are available (equals, disjoint, crosses, ...) but their use depends on the underlying GIS capabilities. Using a query language over GML solves these problems. If a spatial operator is not available at a GIS data source, it can be applied over the GML document obtained from a data source through its WFS server. When executing an external join between two GIS, we only need to join two GML documents retrieved by the two local WFS servers: this is much more comfortable in case of heterogeneous data integration.

When querying GML data, we need to perform spatial analysis (area of a polygon, length of a line) and highlight spatial relationships (containment, overlapping, ...). Languages that are currently used for querying XML data are fully dedicated to tree manipulations, can only get alphanumeric features and are not suitable for spatial calculus. XPath, and more recently XQuery, do not highlight the spatial properties of the set of tags representing Points or Polygons.

The GML tree structure is not suitable for geometric manipulation, and representing geometric data poses structural problems, like the number of tags needed for describing geographic features. This can lead to use GML for data exchange only, but recent work in GIS data integration such as those conducted in the VirGIS project, increased the need of a query language for GML.

Proposals were made for extension to spatial data of existing XML query languages [16], or extension of existing data models to spatial domain [7]. Actually none of them gave birth to a concrete implementation. One solution for querying GML is to come up with an implementation of all spatial functions in XQuery itself, but this is quite difficult, because XQuery is a functional language that doesn't offer data structures for implementing geometric algorithms. Another solution could be the translation of a complete GML document into another language, in order to easily manipulate their properties (for example by the translation, and insertion of a GML document into a database like PostGIS [3]). This could be possible for small documents, but becomes unfeasible with large documents that may contain thousands of features; first of all, because of the translation time, and we also lose the benefit of using XQuery for tree navigation, querying and merging XML datasets. The solution we propose consists in adding spatial operators to XQuery, in order to capture the spatial semantics of a GML document.

2.3 GQuery Data Model

The data model we use is very simple, and is based on XQuery's one (Figure 7). Extensions of another data model for XML [5] or for GML [7] have been proposed; the drawback is that implementation of the database model and the algebra has to be done from scratch. As XQuery is highly recommended by the W3C, we shall stick to the standard.

A GQuery query is composed of expressions. Each expression is made of built-in or user-defined functions. An expression has a value, or generates an error. The result of an expression can be the input of a new one. A value is an ordered sequence of items. An item is a node or an atomic value [18]. There is no distinction between an item and a sequence containing one value.

```

query ::= expression
expression ::= expression ◦ expression | value | ERROR
value ::= (item0, item1, ..., itemi)
item ::= atomic value | node

```

Fig. 7. GQuery Expressions.

There are seven types of nodes: document, element, attribute, text, comment, processing-instruction and namespace nodes.

Writing a query consists of combining simple expression (like atomic values), path expressions (from XPath [18]), FLOWER expression (For-Let-Where-Return), test expressions (if-then-return-else-return), or (pre or user defined) functions. Non spatial operators are arithmetic operators (+, −, ×, /, mod), operators over sequences (concatenation, union, difference), comparison operators (between atomic values, nodes, and sequences), and boolean operators.

Spatial operators are applied to sequences. We have three types of spatial operators, the first two categories perform spatial analysis, the third highlight spatial relationships:

- operators that return numeric values:
 $area, length : sequence = (node) \rightarrow numeric\ value$
 $distance : sequence = (node, node) \rightarrow numeric\ value$
- operators that return GML values: $convexhull, centroid : sequence = (node) \rightarrow node$
- operators that return boolean values:
 $equal, within, touches : sequence = (node, node) \rightarrow boolean$

(where *node* is a GML data node).

Each result of a GQuery expression is part of the data model. The input and output of every query or subexpression within a query is an instance of the data model. GQuery is closed under this query data model. When computing $area(node)$, if *node* is a Polygon, the function returns a numeric value, otherwise it raises an error. In both cases, results are instances of the data model.

Spatial operators can be nested.

For example, $within(convexhull(node1), node2)$ is correct according to the data model. $convexhull(node1)$ is equal to *node*, and $within(node, node2)$ returns a boolean.

3 The VirGIS Mediation System

As mentioned in section 1, one of the motivations behind GQuery is a query model for a geographic integration system. Figure 8 describes the functional architecture of the VirGIS geographic mediation system [11].

The system is mainly composed of three layers: a GIS mediator, Web Feature Servers (WFS) and data sources such as any integration system. Let us note that the WFS servers play the role of wrappers. Indeed they ensure the transformation from the mediator data model to the sources ones and vice-versa. The GIS Mediator is composed of a Mapping Module, a Decomposition/Rewrite module, an Execution module, a Composition module, and a Source Manager module. In this section we present a short description of the different modules of the mediation stage. [10] explains more in details the different steps of query rewriting.

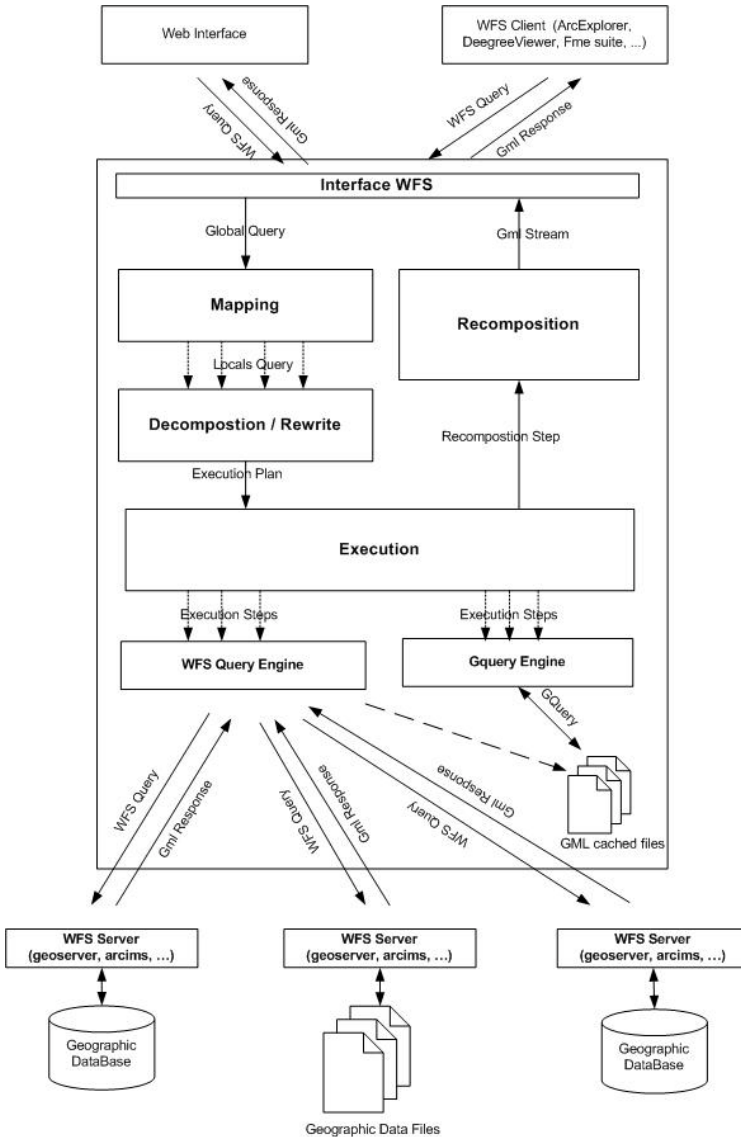


Fig. 8. The VirGIS Mediation Architecture.

The GIS mediator is in charge of analyzing GQuery expressions, including various transformations that involve (meta) schema information, performing some optimizations, and splitting the query into sub-queries, passing them to the right WFS for execution. The WFS layer is in charge of manipulating the data sources: it receives requests from the mediator, executes them, and returns the results. A WFS request consists of a query description or data transformation

operations that are to be applied to one or more features. The request is issued by the client and is posted to a Web server via HTTP. The WFS is invoked to service the request.

A request combines operations defined in the WFS specification. The WFS reads and executes the request and returns an answer to the user as a GML (or XML) document.

Among those operations are:

- GetCapabilities: a WFS source must be able to describe its capabilities. Specifically, it must indicate which feature types it can service and what operations are supported on each feature type.
- DescribeFeatureType: a WFS source must be able, upon request, to describe the structure of any feature it can service.
- GetFeature: a WFS source must be able to service a request, and to retrieve feature instances and properties.

The main components of the GIS mediator are as follows:

1. The WFS translator receives a request that is an XML encoding of (a subset of) a SQL like query language derived from the OpenGIS CQL (Catalog Query Language). This request is translated into a GQuery expression. This module turns the VirGIS system into a WFS-enabled server.
2. The Mapping module uses integrated schema (meta) information in order to express user queries in terms of local source schemas. Each mapping rule express a correspondence between global schema features and local ones. Those correspondences are expressed in using of path-to-path mappings as it is explained in [10]. Note that conditions and aggregations (like topological operators) are used to describe local sources contents and capabilities.
3. The Decomposition/Rewrite module exploits information about source feature types and source capabilities to generate an execution plan. A global GQuery expression is used as a container (place-holder) for collecting and integrating results coming from local data sources. The rewriting algorithm [10] is inspired from the one used in the Styx system [4].
4. The Execution module processes sub-queries contained in the execution plan it receives in, sending them to the appropriate WFS. Note that sub-queries (according to their type) are executed either by the WFS Query engine or the GQuery engine. For example, queries that require an operator that is not available at any of the integrated data sources is processed by the GQuery engine.
5. The Composition module treats the final answer to delete the duplicated answers, etc. it produces a GML document which is returned to the client.
6. The Source Manager module is in charge of collecting information from the WFS sources. It builds the configuration files for the integrated (Global) schema and sources capabilities information.

4 Illustrating Example

4.1 Application Scenario

The scenario involves the integration of satellite images catalogues. As an example, consider query Q below:

Given a spatial location (e.g., Corsica) return all available satellite shots.

Figure 9 below illustrates a subset of schemas drawn from SPOT and IKONOS catalogues, and the VIRGIS mediated schema. QUICK_LOOK table refers to a sample of small images called *quick looks* that give an overview of satellite images supplied in the catalogue. The global query Q is posed against the VIRGIS relation. This query does not specify any satellite, hence information should be retrieved from all integrated satellites. Note that we could also have specified a satellite name (e.g., SPOT or IKONOS).

Attribute	Type	Attribute	Type	Attribute	Type
date_	Date	date_acqui	Date	key	string
sun_elev	numeric	sun_el	numeric	filename	string
satellite	string	satellite	string		
sat_id	numeric	sat_id	numeric		
key	string	key	string		
the-geom	Polygon	the-geom	Polygon		

SPOT
IKONOS
QUICK LOOK

```
VIRGIS(id: string, date: date, sun_elevation: numeric,
       name:string, satid: string, url: string, geom: Polygon)
```

Fig. 9. Locals and Global Satellite Schemas.

Normally processing query Q is time consuming, may require a tremendous effort, and is practically unfeasible:

- Data sets collected by satellites are quite large and it is impractical for a user to examine the complete data sets.
- Each catalogue has its own organization and ontology, which leads to well known semantic interoperability problems.
- A user has usually access to only one catalogue at a time, generally in using a browsing system, because there is no integrated system that provides (transparent) access to multiple catalogues.

4.2 A Simple Visual Query Interface

Because neither GQuery nor WFS expressions are easy to express for a naive user, we designed a high level friendly user interface that allows simple queries.

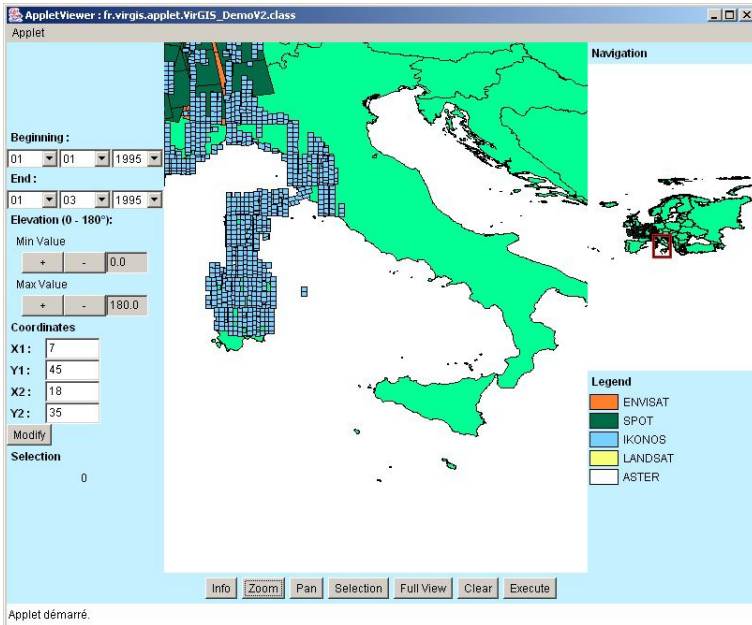


Fig. 10. VirGIS Query Interface.

As illustrated in Figure 10, a user query is very intuitive: users graphically choose the region they are interested in, they specify some values such as dates (begin, end), sun elevation, etc.

4.3 Query Processing

In this section, we describe the 3-phase processing of query Q mentioned above. The first phase consists in mapping the global WFS query to local virtual WFS queries, which led to the following WFS expression represented in Figure 11.

```
<GetFeature xmlns:gml="http://www.opengis.net/gml">
  <Query typeName="satellite">
    <Filter>
      <BBOX>
        <PropertyName>geom</PropertyName>
        <gml:Box><gml:coordinates decimal="." cs="," ts=" "> 8,43 10,41
          </gml:coordinates>
        </gml:Box>
      </BBOX>
    </Filter>
  </Query>
</GetFeature>
```

Fig. 11. The WFS (Global) Query.

The second phase is performed by the *Decomposition/Rewrite* module that is in charge of generating an execution plan. The query plan consists of both WFS and GQuery subqueries together with their scheduling information represented by a *Priority* timestamp attribute.

Figure 12 (resp. 13) below illustrates a WFS (resp. GQuery) expression being part of the execution plan.

In the third phase, WFS queries are sent to data sources, while GQuery expressions are executed by the GQuery processor. WFS queries result in GML streams that may be passed to GQuery queries, or considered as partial GML results. For example, document `temp33.xml` returned by a WFS server is passed to GQuery request represented in Figure 13. Finally, all GML portions are assembled and recomposed in order to generate a final GML document, that is the final answer to be returned to the user.

```
<?xml version="1.0" encoding="UTF-8"?> <GetFeature
  xmlns:gml="http://www.opengis.net/gml">
  <Query featureSourceId="Geoserver_local_preview"
    queryIndex="0" typeName="satellite">
    <PropertyName featureSourceId="Geoserver_local_preview"> id
    </PropertyName>
    <PropertyName featureSourceId="Geoserver_local_preview"> url
    </PropertyName>
  </Query>
</GetFeature>
```

Fig. 12. A WFS Subquery.

```
<!-- GQuery query, Priority = 1, Output = temp35.xml-->
<wfs:FeatureCollection>
{
  for $x1 in document("temp31.xml")/wfs:FeatureCollection
    /gml:FeatureMember/satellite,
    $x2 in document("temp33.xml")/wfs:FeatureCollection
    /gml:FeatureMember/satellite
  where $x1/id=$x2/id
  return
  <gml:FeatureMember>
    <satellite fid={$x1/@fid}>
      {$x1/date}{$x1/elevation}{$x1/geom}
      {$x1/id}{$x1/name}{$x1/satid}{$x2/url}
    </satellite>
  </gml:FeatureMember>
}
</wfs:FeatureCollection>
```

Fig. 13. A GQuery Subquery.

5 Concluding Remarks

With the proliferation of GIS data and resources over the Internet, there exists a huge demand for robust geospatial information services that allow federation/interoperation of massive repositories of heterogeneous data and metadata.

To tackle this problem, we developed an XML (GML) based integration system called VirGIS. The technical choices we made address effective integration needs expressed by the GIS community. We believe that we do have a piece of software that may be used as a basis for online Geoservices. Indeed, since VirGIS accepts OGC's WFS queries, it may deliver any combination of Web Feature Servers.

Future research should include query optimization and automatic schema matching.

Acknowledgement

The work presented in the paper was partially supported by a grant of the French Ministère de l'éducation nationale, de la recherche et de la technologie in the context of the programme Réseau National de recherche et d'innovation en Technologies Logicielles (RNTL).

References

1. Esri interoperability and standards. <http://www.esri.com/software/opengis>.
2. Intergraph. <http://www.intergraph.com>.
3. PostGIS, Geographic objects for PostgreSQL. <http://postgis.refractory.net/>.
4. Bernd Amann, Catriel Beerli, Irini Fundulaki, and Michel Scholl. Querying XML Sources Using an Ontology-Based Mediator. In *On the Move to Meaningful Internet Systems, 2002 – DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 429–448. Springer-Verlag, 2002.
5. David Beech, Ashok Malhotra, and Michael Rys. A formal data model and algebra for XML. <http://www-db.stanford.edu/dbseminar/Archive/FallY99/malhotra-slides/malhotra.pdf>, 1999.
6. Omar Boucelma, Zoé Lacroix, and Mehdi ESSID. A WFS-Based Mediation System for GIS Interoperability. pages 23–28. ACM GIS'02, November 2002.
7. J.E. Corcoles and P. Gonzalez. A Specification of a Spatial Query Language over GML. pages 112–117. ACM GIS'01, November 2002.
8. M.J. Egenhofer. Spatial SQL: a Query and Presentation Language. *IEEE TKDE*, 6(1):86–95, November 1994.
9. F. Wang and J. Sha and H. Chen and S. Yang. GeoSQL: a Spatial Query Language for Object-Oriented GIS. *CSIT'2000*, 2000.
10. M. ESSID, O Boucelma, Y. Lassoued and F.M. Colonna. Query Processing in a Geographic Mediation System. In *Proceedings of The 12th International Symposium of ACM GIS*, Washington D.C, November 2004.
11. Omar Boucelma and Mehdi ESSID and Zoé Lacroix and Julien Vinel and Jean-Yves Garinet and A. Betari. VirGIS: Mediation for Geographical Information Systems. In *ICDE'04*, Boston, USA, March 30 – April 2, 2004.

12. OpenGIS. Geography Markup Language (GML) 2.0 – Document 01-029. <http://www.opengeospatial.org/>, February 2001.
13. OpenGIS. Geography Markup Language (GML) 3.0 – Document 02-023r4. <http://www.opengeospatial.org/>, January 2003.
14. OpenGIS. OGC Request 13: OpenGIS Web Feature Server Implementation Specification, 2001. see <http://www.opengis.org/info/techno/rfp13info.htm>.
15. OpenGIS Consortium. OpenGIS specifications. <http://www.opengeospatial.org/>, 2003.
16. Ranga Raju Vatsavai. GML-QL: A Spatial Query Language Specification for GML. *UCGIS Summer 2002, Athens, Georgia*, 2002.
17. Ph. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases – with applications to GIS*. Morgan Kaufmann, 2001.
18. World Wide Web Consortium. W3c specifications. <http://www.w3c.org>, 2003.

A Generic Framework for GIS Applications^{*}

Miguel R. Luaces, Nieves R. Brisaboa, José R. Paramá, and Jose R. Viqueira

Laboratorio de Bases de Datos
Facultade de Informatica, Universidade da Coruña
Campus de Elviña, s/n. 15071, A Coruña, Spain
+34 981 16 70 00 Ext. 1306
Fax: +34 981 16 71 60
{luaces,brisaboa,parama,joserios}@udc.es

Abstract. Geographic information systems (GIS) are becoming more usual due to the improved performance of computer systems. GIS applications are being developed using the three-tier software architecture traditionally used for general-purpose information systems. Even though this architecture is suitable for GIS applications, the special nature and exclusive characteristics of geographic information pose special functional requirements on the architecture in terms of conceptual and logical models, data structures, access methods, analysis techniques, or visualization procedures.

In this paper, we propose a generic architecture for GIS that provides support for the special nature of geographic information and conforms with the specifications proposed by the ISO/TC 211 and the OGC. Our strategy to achieve this goal consists of two steps: (i) we analyze the special characteristics of GIS with respect to traditional information systems, (ii) and we adapt the traditional three-tier architecture for information systems to take into account the special characteristics of GIS.

Finally, we have tried to apply the architecture that we propose in the development of a complete and complex real-life GIS application using commercial tools in the analysis, design and implementation. We describe this application, and we use it to describe the limitations of current commercial GIS development tools by analyzing the differences in the architecture of the resulting system with respect to our proposal.

1 Introduction

Until a few decades ago, manipulating, synthesizing and representing geographic information was restricted to paper maps and these tasks were limited to manual, non-interactive processes. The exponential improvement in the performance of computer-based technologies and the increasing demand for interactive manipulation and analysis of geographic information have created a need for *geographic information systems* (GIS) [1, 2]. An important characteristic of geographic information systems is that they are more than tools to produce paper maps.

^{*} This work was partially granted by CICYT (refs. TIC2003-06593 and FIT-150500-2003-588), and Xunta de Galicia (ref. PGIDIT02SIN10501PR).

Whereas in traditional cartography the paper map is the database, in a GIS the map is only a projection of a particular view of a geographic database at a given time. This enables the GIS end-user to review an unlimited number of analysis alternatives, and to make maps from different points of view emphasizing different aspects of the information. As a consequence, functional requirements for GIS are vast and go far beyond those of traditional information systems [3–6].

After many years of research and development, it is now generally accepted that the architecture of general-purpose information systems must consist of three separate tiers, namely: the *presentation tier*, the *application logic tier* (or *business logic tier*), and the *data tier*. The main advantage of this architecture is that it enforces a strict separation of the functionality of the system into three different independent modules that interact only at well-defined interfaces. This enables a developer to modify each one of these modules of the application with little impact on the others. Therefore, this architecture provides increased performance, flexibility, maintainability, reusability and scalability.

Even though the three-tier architecture for general-purpose information systems is suitable for GIS, the special nature and exclusive characteristics of geographic information pose special functional requirements on the architecture in terms of conceptual and logical models, data structures, access methods, analysis techniques, or visualization procedures. For instance:

- Special data types and operations are needed to represent and manipulate geographic information.
- Geographic information requires many different analysis and visualization procedures.
- Geographic information is typically voluminous with a naturally imposed hierarchical structure.
- Geographic information processing is characterized by transactions that are much longer than a typical standard relational database transaction.
- There are two different conceptual views of geographic space: an object-based view and a field-based view.
- Additionally, each conceptual view of space can be represented in many different ways in a computer.

These and other features impact the overall architecture of a GIS. Therefore, it is very important to determine the special requirements and functionality of GIS applications beyond those of general-purpose information systems in order to design and implement a GIS application framework with appropriate capabilities for modeling, collecting, querying, and visualizing geographic information. This is precisely the main goal of our work: proposing a generic architecture for a GIS framework that provides support for the special characteristics and requirements of geographic information.

The rest of this paper is structured as follows. First, in Section 2 we analyze the special features of geographic information that pose requirements on the architecture of GIS beyond those common to general-purpose information systems. After that, in Section 3, we review the OpenGIS Consortium (OGC) and the ISO Technical Committee 211 (*ISO/TC 211, Geographic Information/Geomatics*)

proposals of standards for representing and manipulating geographic information. We also point out in this review the benefits and drawbacks of these proposals. Then, in Section 4 we introduce our proposal for a generic architecture for GIS that fulfills the requirements described in Section 2. Section 5 describes the development of a GIS for the Provincial Council of A Coruña that tries to apply our proposal, and Section 6 is devoted to the analysis of the differences between the proposed architecture and the implemented GIS application caused by the limitations of commercial GIS development tools. We end this work by giving some concluding remarks and describing future work in Section 7.

2 Special Characteristics of Geographic Information

The special nature of geographic information imposes some requirements on the architecture of the information system. We enumerate some of these requirements in this section.

Representation of Geographic Information. The conceptual models used for information systems (e.g., the entity-relationship model) do not have constructs to model application schemas that deal with geographic information. Furthermore, logical models (e.g., the relational model) are strongly geared toward business applications that manipulate large but simple data sets, and do not include functionality to represent geographic information. In addition to data types for numbers, texts and dates, the representation of geographic information on a computer requires new data types such as *point*, *curve*, *surface*, or collection types like *point collection*, or *geometry collection*. Finally, physical models for traditional information systems are unable to represent efficiently geographic information. Summarizing, traditional information systems must be extended at all levels from the conceptual model to the physical model in order to represent geographic information adequately.

Geographic Information Processing. There is a rich set of special transformation, manipulation and analysis techniques applicable to geographic information, which must be integrated within the information system. This must be done by providing an exhaustive set of primitive operations on the data abstractions of the conceptual model, which must be integrated in a query language to retrieve and manipulate the data abstractions of the conceptual model. Finally, problem-solving techniques must be used for the following categories of problems:

- *What is it at this location?, and Where is this located?*
- *What is the spatial relationship between these objects?*
- *What location satisfies these requirements?*
- *What will be the situation in the future?*

Visualization of Geographic Information. The visualization of geographic information is a distinctive feature of GIS applications compared with general-purpose information systems. Geographic information has some peculiarities that have an impact on the presentation process:

- Different abstractions must be used for the representation and the visualization of the information.
- It is multi-dimensional.
- It is voluminous.
- It is required at varying scales and sizes.
- It is required from different perspectives.
- It is projected onto a flat surface (i.e., on a computer screen or a paper).

Therefore, there is a need for appropriate metaphors to manipulate geographic information at the user interface of the system. These metaphors must be based on the well-known map metaphor, and must incorporate dynamic operations such as *zoom*, *pan* and the addition and removal of information.

System Architecture. The special nature of geographic information makes more important the fulfillment of some requirements of general-purpose information systems, such as flexibility, extensibility, reusability, scalability, reliability, and security. In order to provide these features, the architecture of the GIS must be based on an extensible DBMS providing geographic information management services, and a collection of modular, highly-distributed, geographic information processing and visualization services.

Other Issues. Geographic information poses other important requirements on the architecture of the system. First, as any other application domain for information system, the field of GIS needs special metadata elements to describe the particularities of the application domain. Metadata in the architecture of a GIS enables to find geographic information data sets, to describe how the information can be used, and to check whether the information satisfies some requirements.

In addition to this, the amount of possible analysis techniques for geographic information is unlimited. Therefore, no GIS development tool can provide all possible operations and problem-solving techniques. As a consequence, the architecture of a GIS development tool must be *extensible* to support the addition of new data analysis procedures. The process of extending the functionality of a GIS development tool is usually referred to as *customization*.

Finally, the temporal component of geographic information was systematically ignored by GIS development tools and the more general research field of spatial databases. This is changing in the last years, and many research efforts has been dedicated to the emerging field of *spatio-temporal* and *moving object* databases [7].

3 International Standards for Geographic Information Systems

Given that each application has a different point of view on geographic information, each developer has defined conceptual models, geographic data models, storage formats, analysis operations, or representation procedures specially

adapted to the requirements of the application. As a consequence, there is nowadays a problem of interoperability between the GIS development tools; it is not easy to use one tool to analyze the information collected with another tool.

In order to solve this problem, a number of government, research and industry partners founded the OpenGIS Consortium (OGC) in 1994 to promote interoperability among GIS development tools [8]. Another standards organization that has devoted many efforts to GIS applications is the International Organization for Standardization (ISO) by means of the ISO Technical Committee 211 (ISO/TC 211), named *Geographic Information/Geomatics* [9]. The purpose of both organizations is to create specifications of standards concerning geographic information with detail enough to enable developers to create implementations conforming to these standards that interoperate without problems.

For many years, the OGC and the ISO were working independently to reach overlapping goals, but nowadays both bodies seek to converge toward a common solution. The OGC and the ISO/TC 211 are carrying out a very important task in the field of geographic information systems. They are laying the foundation for a new generation of GIS applications and development tools that will be able to cooperate to a greater extent at many different levels. Furthermore, instead of proposing a monolithic software layer implementing all the functionality of GIS applications, these organizations are proposing to break down the functionality in a vast collection of services with very specific functionality that interact only at the interfaces. This allows a very flexible GIS application architecture because the services implementing the functionality may be running in a single computer, or distributed along a wide-area network, in a totally transparent way.

We have depicted in Figure 1 the architecture of a GIS application with two different user interfaces built over the specifications already published. The first interface consists in a desktop-based GIS for geographic data analysis, and the second one is a web-based GIS for geographic information portrayal. The figure shows the specifications that are already adopted in white boxes, and the missing pieces of the architecture in gray boxes.

The work developed by these organizations has also some little drawbacks. For instance, even though the specifications provide a complete information management tier, much of the processing algorithms and information portrayal techniques must be defined by a developer. No specifications have been defined for the processing tier, and very little work has been produced for the human interface tier.

Another little problem is that the intrinsic nature of any standards organization causes the process of developing a specification to be rather slow. Before an specification is adopted, it must be proposed, written in a draft state, discussed and voted by the membership. This is a lengthy process that cannot be assumed by software companies that must produce novel products at a much faster pace.

Finally, the definition of specifications by multiple groups working independently may cause that the specifications do not match perfectly due to coordination problems. Some concrete examples of this problem are shown in the following list:

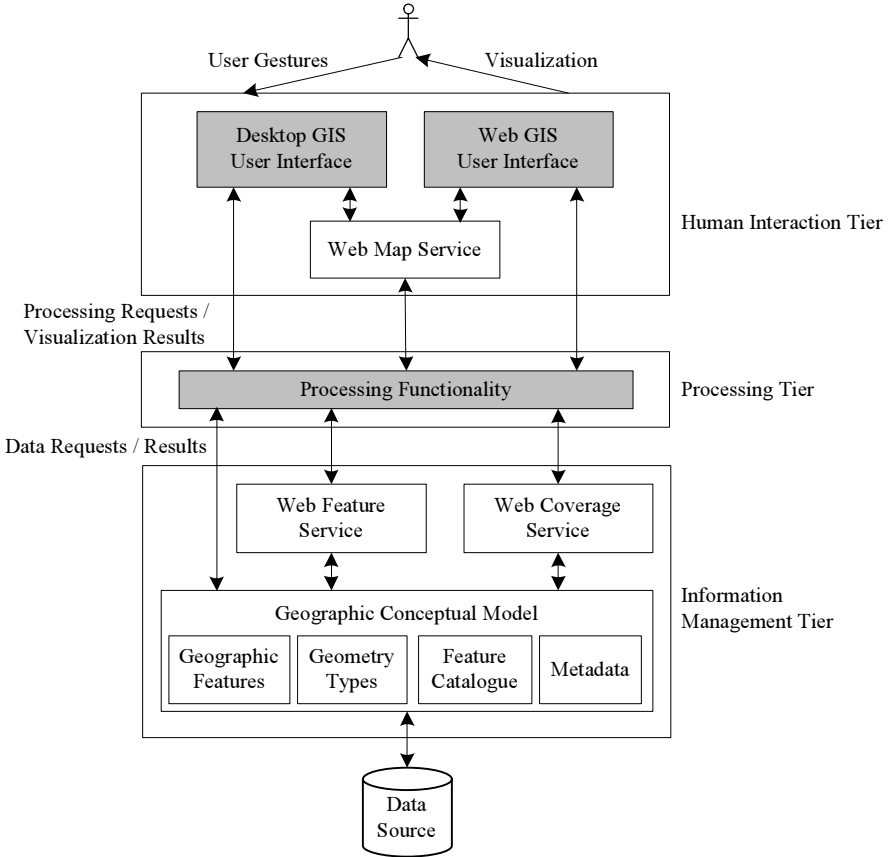


Fig. 1. OGC Services in the Architecture of a GIS.

- *Duplicate definitions.* The specifications for both the Geographic Markup Language (GML) [10] and the Styled Layer Descriptor language (SLD) [11] provide a definition for constructs to specify the visual style of a cartographic object generated from a geographic feature.
- *Missing functionality.* The Web Feature Service (WFS) [12] is defined as a service for querying and manipulating geographic features. However, instead of providing a complete query language like the one defined for the simple feature geometry model [13], the definition imposes some limitations on the types of queries that can be posed over the data source (e.g., the language does not support relational joins, new values in the query result cannot be computed, or the complete set of spatial operations is not available).

In spite of these little drawbacks, the architecture and the specifications proposed by these organization must be considered as a starting point for the development of any GIS application. These specifications are the greatest source inspiration for the architecture described in the following section.

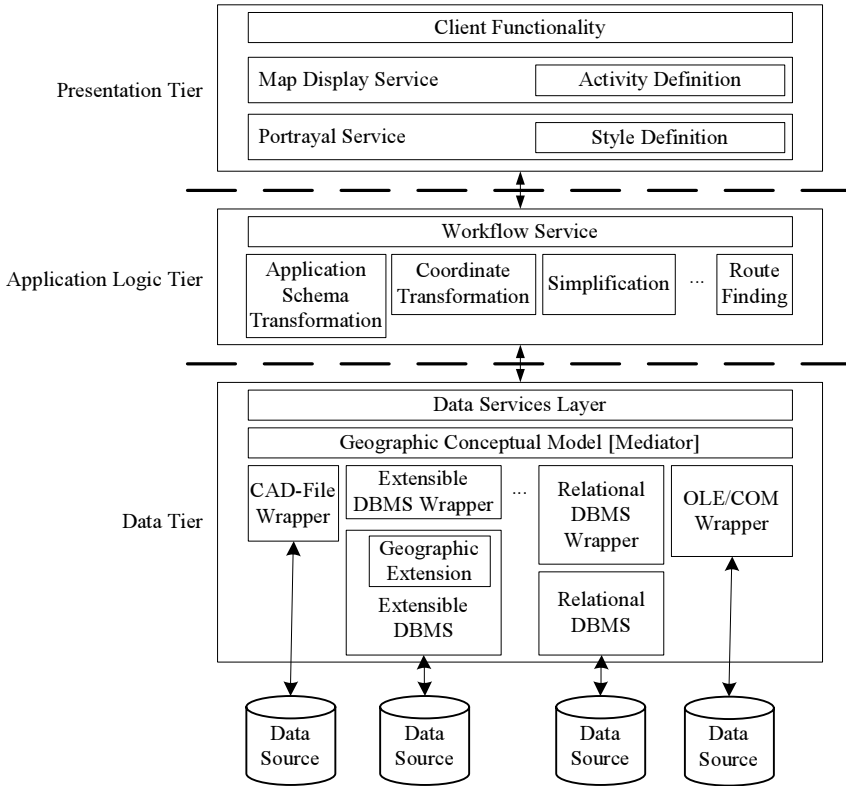


Fig. 2. A Generic System Architecture for GIS.

4 A Generic Architecture for Geographic Information Systems

After analyzing the special characteristics of geographic information and the requirements that they pose over the architecture of geographic information systems, we present in this section a description of the components and the interactions in a generic architecture for geographic information systems that meets these requirements. The design of this architecture is heavily influenced by the proposals of the ISO/TC 211 and OGC, and reuses the work of these organizations where their specifications are mature.

Figure 2 shows our proposal of a generic architecture for geographic information systems. The architecture separates the functionality of the system in three independent tiers, namely the *Data Tier*, the *Application Logic Tier* and the *Presentation Tier*. The Data Tier provides data management functionality independently from the software technology. The Presentation Tier is responsible for implementing the user interface of the system, displaying the maps and providing some basic functionality over them. Finally, the Application Logic Tier implements the problem-solving functionality of the system.

In order to enable reusability and flexibility of the system architecture, the functionality of these tiers must be implemented independently of any particular application. The strategy followed consists in finding and isolating the characteristics that are independent from the application schema and functionality from the dependent ones. Then, the independent characteristics of the system are implemented once using generic algorithms. The architecture built with these modules can be used as a framework for GIS applications by filling in the specific details of the application. We describe now each layer in more detail.

4.1 Data Tier

The purpose of the data tier is to provide information management functionality independently from the software technology used to store the data. This tier receives information retrieval and manipulation requests expressed using a query language, evaluates the query, and returns a set of data objects that are represented using an information exchange language.

Considering that there may be many different types of data sources, the internal architecture of the tier must be organized in a mediator-wrapper pattern. The mediator layer consists in a conceptual model for geographic information including data types and a query language for representing and manipulating geographic information, and metadata and catalogue information. Then, a wrapper module must be implemented for each different type of data source. For an extensible DBMS that supports the conceptual model directly by means of an extension module, the wrapper is very simple. On the other hand, for a relational DBMS the wrapper modules uses tables or large objects to store geographic information and implements the query language using memory operations. Similarly, legacy data formats like CAD files must be supported by wrapper modules.

The query language can be a text-based high-level query language (e.g., SQL derivatives), or it can be a low-level query language based on an application programming interface. The same alternatives exist for the query results, which may be accessed using a high-level, text-based descriptive language (e.g. GML [10]), or a low-level, application programming interface (e.g. OpenGIS Simple Features for OLE/COM [14]).

The topmost interface of this tier consists of a collection of *data services* that provide profiles of the conceptual model for specific applications (i.e., subsets of the conceptual model, query and results language, and metadata facilities). These services enable to hide the complexity of the underlying conceptual model by providing, for example, a simplified query language or exchange language. Two examples of services that can be implemented in this layer are the Web Feature Service and the Web Coverage Service defined by the OpenGIS.

4.2 Application Logic Tier

The application logic tier comprises the business logic of the system. This tier must be composed of multiple independent *services*, which are modules responsible of performing well-defined and simple tasks. Each service is defined by giving

its interface as a set of operations and a description of the results. When a service operation is invoked, the service module answers the request either by using its internal information, or by building and issuing the appropriate queries to the Data Tier and manipulating the data returned.

The services existing in the architecture cannot be predefined because the functionality necessary for GIS applications cannot be known in advance. We can foresee the need for some services such as a route-finding service in networks, a geographic value simplification service, a coordinate transformation service, or an application schema transformation service. In order to achieve more complex tasks, multiple services must be chained by means of a *workflow service*. This kind of service allows developers and end-users to build a new service by connecting a collection of simple services.

4.3 Presentation Tier

The Presentation Tier is responsible for the user interface of the system that enables data visualization, data manipulation and data entry. The presentation tier receives the user interaction in the form of mouse gestures, keyboard inputs or other device inputs. These inputs are evaluated and the appropriate operations in the application logic tier are invoked. When the results are returned, they are displayed to the user using the appropriate user interface controls and visualization metaphors.

The most important component of the presentation tier is the portrayal service, which is in charge of converting a collection of geographic features into a collection of cartographic objects that can be rendered on a display device. The portrayal process is controlled by a set of style definitions, which must define precisely the way in which each geographic feature must be rendered.

The resulting cartographic objects are visualized using a *map display* service, which uses a *map metaphor* to allow the end-user to manipulate the displayed map. For instance, the map scale is changed by using the *zoom-in* and *zoom-out* metaphors.

In addition to visualization manipulations (e.g., scale and view change), the map display component must allow the end-user to manipulate the cartographic objects displayed in the map to perform geographic operations and to request processing operations from the application logic tier. The *activity module* associates these actions to user interface events that occur in the map. As an example, a developer can associate to the event *click over an element in the map* the action *display element information*. The implementation of the action is responsibility of the developer. However, it is necessary that the operations of the features in the Data Tier are accessible to the developer.

If all the functionality of the system is implemented in the application logic tier, the response time of the system may not be fast enough. In order to achieve faster response times from the system, the presentation tier may implement some functionality using the cartographic objects for the computations. This is implemented in the *presentation functionality* module, and consists in tools for the following tasks:

- *Control of the map scale and position.* This allows the end-user to focus on the map phenomena of interest by zooming and moving the map.
- *Management of the graphical legend.* It allows to understand the map by describing the real-world entities that are described by each symbol. It also allows to the end-user to customize the map by adding and removing additional features.
- *Measure distances and areas.*
- *Display the map context.* For instance, using an overview map, a graphical scale bar, or a north arrow.

4.4 Summary

We have proposed in this section a generic architecture for GIS applications and we have also described briefly each component of the architecture. This architecture proposal provides support for the special characteristics of geographic information and conforms with the specifications defined by the ISO/TC 211 and OGC where possible.

5 The EIEL Geographic Information System

In order to discover the funding needs of each municipality and to propose special action programs to balance the living conditions of the municipalities, each provincial council in Spain is required to conduct, every five years, a survey on local infrastructure and facilities, (named EIEL from the Spanish *Encuesta de Infraestructura y Equipamientos Locales*). The amount of information collected by the survey demands a tool to objectively analyze and evaluate the situation and state of infrastructure and facilities in each municipality.

The province of A Coruña is located in northwestern Spain. With more than one million inhabitants and almost eight thousand square kilometers, it is densely populated with more than a hundred and twenty-five inhabitants per square kilometer. The provincial council of A Coruña decided to broaden the goals of the EIEL for the year 2000. More particularly, these new goals were considered:

- Extend the information to be collected, both in terms of the different kinds of elements to be surveyed, and the amount of information for each particular item.
- Reference the items surveyed to its geographical location or extent.
- Build an information system with the information collected to be used by the provincial council staff, and build a publicly-accessible, web-based information system.

These goals were achieved through a two-year project carried out by the University of A Coruña. A large group of students from the civil engineering school and the architecture school, supervised by a group of professors, collected the data by direct observation or interviewing the responsible staff in each municipality. At the database laboratory of the University of A Coruña, we designed

and developed the applications supporting the data collection work flow. Then, we developed a geographic information system to manage and exploit this information [15].

We tried to apply the architecture proposed in Section 4 for the implementation of this GIS. This enables us to prove the applicability of our proposal in a real-world problem. Moreover, since we wanted the system to be ready in as little time as possible, we decided to use existing commercial applications instead of developing new software components from scratch.

Two independent applications were built at the end of the project:

- *A data maintenance tool (GISEIEL)*. We designed and implemented an application to enable the responsible staff at the Provincial Council to correct and update the information stored in the GIS.
- *A web-based data exploitation tool (WebEIEL)*. This application was designed in order to enable all the staff at the Provincial Council and citizens to browse the information collected by the EIEL. It can be found at <http://www.dicoruna.es/webeiel/>.

Figure 3 shows the system architecture of *WebEIEL*. The components that we had to develop are shown with a gray background, whereas the commercial components are shown with a white background. On the server side, the data is managed by a relational DBMS (Microsoft SQL Server 7.0) and the geographic

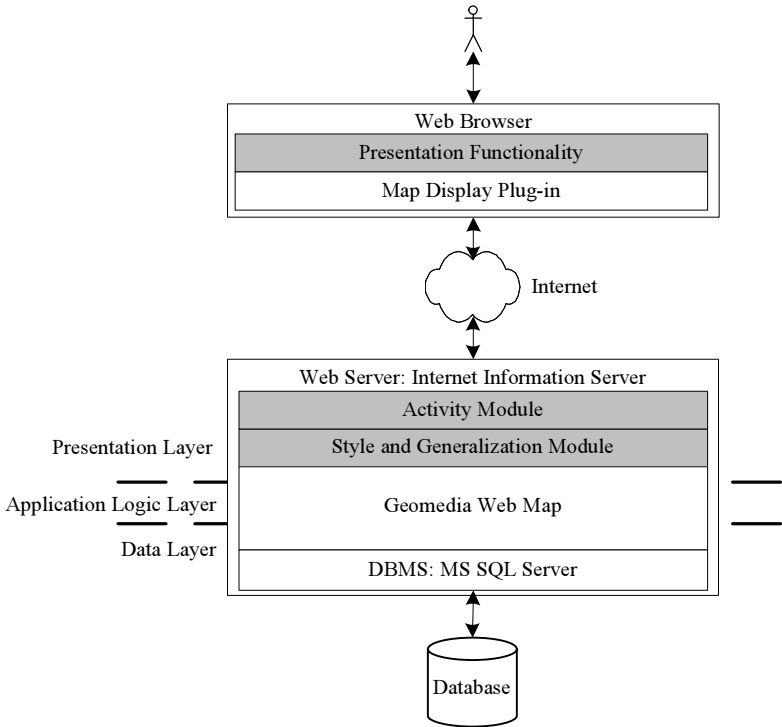


Fig. 3. System Architecture for WebEIEL.

data model is provided by Intergraph Geomedia Web Map. The information of the web application is served by Microsoft Internet Information Server, the web server supplied by Microsoft. This particular server was chosen because it is required by Geomedia Web Map.

The client side of *WebEIEL* was implemented using dynamic HTML and a map display plug-in provided by Intergraph. This plug-in consists in a Java applet that displays the map information implements many presentation functionality such as the computation of distances and areas. The main reason for choosing a *thick-client* approach for the client was two-fold. First, to provide some interactive functionality in the client such as highlighting selected geographic objects, or displaying a tooltip message when the mouse is over a geographic object. And second, to display the map information using a vector image format that provides a much higher quality than raster image formats.

We had to develop many new modules and customize many existing components in order to implement this system, because the ones provided by Intergraph Geomedia Web Map were not sufficient to meet our requirements. Particularly, we implemented:

- *Style and generalization module.* One of the requirements of our application was that the information displayed in the client should be in a vector format in order to produce high-quality maps. Moreover, the map display plug-in enables enhanced user interactions (e.g., object highlighting, tooltips) if a vector format is used for the map. However, the geographic information collected by the survey was very detailed, and the maps produced directly with this information were very large and were not suitable for web-based visualization due to long transmission times and complex rendering. In order to solve this problem, we implemented a module to perform automatic generalization of geographic information. This module uses different representations of the same geographic value at different detail levels, and a set of rules that determine which geographic value must be used for a given map scale. This enables to reduce the size of a map by simplifying geographic values at small map scales. Additionally, this module also enables a developer to easily define map layers, styles, and complete maps.
- *Activity module.* In addition to a definition of the map contents, a developer must also provide a definition of the actions to be performed in response to user interactions. For instance, it is necessary to define the action associated to a user mouse click on a geographic value. We developed a module to facilitate the definition, management and implementation of this functionality.
- *Presentation functionality.* Even though we used a *thick-client* in the client side of the application, we had to provide some client-side presentation functionality such as presentation of alphanumeric information for geographic values, or the management of the graphical legend of the map.

A screen capture of the web-based application is displayed in Figure 4. It shows the graphical legend management on the left, the tool bar on the top, the context information area on the bottom, and the map in the center of the image. A window with alphanumeric information of the selected element is shown overlaid.

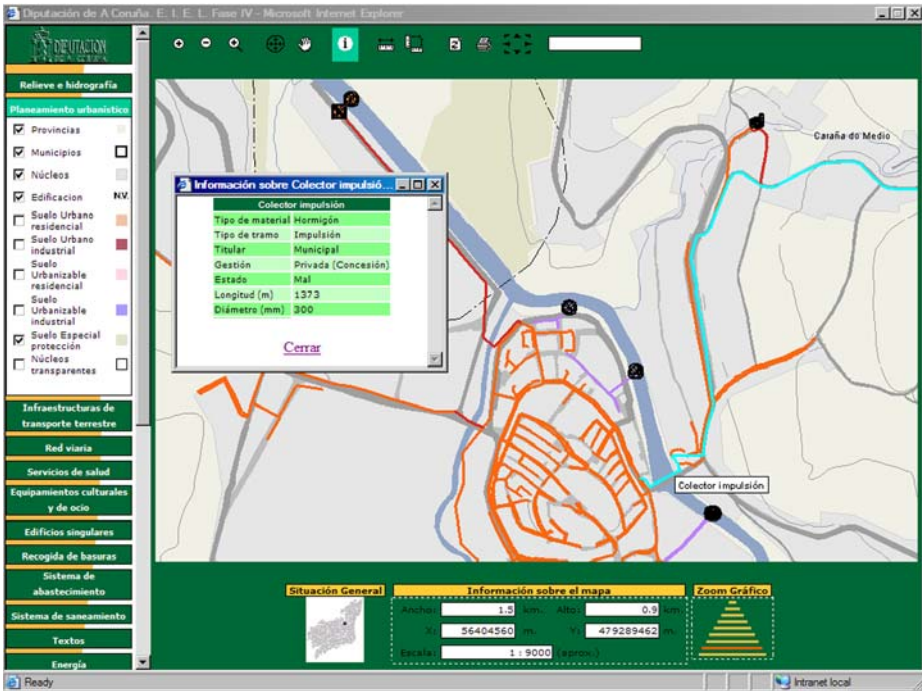


Fig. 4. User Interface of WebEIEL.

6 Analysis of the WebEIEL System

As we already said, in order to reduce the development time, we used commercially-available tools instead of custom-developed software modules in the architecture of the *WebEIEL* application wherever it was possible. This caused the resulting system to be different from the architecture we propose in Section 4. These differences are common to almost any GIS application developed using a commercial GIS development tools. They can be categorized as follows:

- *Incorrect functionality separation.* In the architecture we propose, we have presented a sound division of the functionality into independent software tiers. Moreover, the functionality of these tiers can be further divided into independent software layers. However, in the architecture of the GIS described in Section 5, this principle does not hold. Figure 3 shows the separation of the system architecture using thick dashed lines. It can be seen that Intergraph Geomedia Web Map provides in a monolithic software module functionality for the three tiers of the architecture. This is a common problem in many commercial GIS development tools that causes many problems regarding flexibility, scalability, and reusability.
- *Missing functionality.* Another common problem in GIS development tools is that some important functionality described in Section 4 is missing. In

our case, the conceptual model for geographic information does not support multiple geographic values in a table.

Furthermore, the conceptual data model was not implemented using an extensible DBMS. Instead, a relational DBMS had to be used and the geographic values were stored using BLOBs, which caused many efficiency problems. For instance, the computation of a spatial join requires to load both relations in main memory, and perform a selection using main-memory join algorithms.

Finally, the conceptual data model does not support the declaration of explicit topological relationships. This leads to errors in the geographic information such as error as *gaps* between geographic values that share a border.

- *Technology independence.* GIS development tools often impose a proprietary data storage format, and proprietary interfaces to the data tier. Sometimes, a proprietary data presentation format is also imposed on the system. The result is that the application is too heavily integrated with the technology and cannot be ported to other software platforms.

7 Conclusions and Future Work

Geographic information is slowly becoming an important element in computer systems. Many applications are being developed for industrial, administrative and research tasks in which geographic information is the central component. Furthermore, geographic information is providing added-value to many applications that did not consider it before (e.g., location-based services).

However, geographic information is a special kind of information that cannot be represented, manipulated and visualized using the methods that were traditionally used for other business and scientific information. Geographic information requires special modeling and analysis methods. The first contribution of our work is an analysis of the special characteristics that make geographic information system different from traditional information systems.

Furthermore, here does not exist a generic architecture for GIS that takes into account the special nature and characteristics of geographic information and at the same time the well-known requirements for general-purpose information systems. The OpenGIS Consortium and the ISO are working on specifications of standards concerning geographic information. They are laying the foundation for a new generation of GIS applications and development tools that will be able to cooperate to a greater extent at many different levels. The main contribution of our work is a proposal for a generic architecture whose design is based on the analysis of the special characteristics of geographic information with respect to traditional information. This architecture is based on the ISO/TC 211 and OGC proposals, and conforms with their specifications where possible.

The solutions provided by commercial tools do not completely satisfy the requirements derived from the special characteristics of geographic information and the requirements of general-purpose information systems. We have presented the complete development process of a complex GIS for the Provincial Council

of A Coruña, and we have compared and analyzed the differences between the architecture we propose and the completed GIS. Commercial systems cause serious efficiency problems and lack of functionality in some cases. Nevertheless, new commercial tools such as Oracle Spatial Option or ESRI Spatial Data Extender are moving towards the right direction by implementing international standards for representing geographic information, and providing a well-structured solution for the functionality they provide.

We have shown in the architecture that there is much functionality that is independent of the particular application of the GIS. This functionality can be implemented in a generic manner, and adapted later on with application-specific details given using high-level languages. However, current commercial GIS development tools do not provide this functionality, and therefore it must be implemented *ad hoc* using programming languages.

The next steps suggested by our work are:

- *Migrate the EIEL GIS to Oracle Spatial Option.* Oracle Spatial Option provides a correct implementation of the functionality proposed for our architecture in the sense that the representation and manipulation of geographic information is performed within the DBMS. In order to analyze the improvement in performance resulting from this feature, we are currently migrating the EIEL GIS to Oracle Spatial Option.
- *Implement the generic modules of the architecture.* Even though we have already implemented some of the generic modules described in the architecture for the project described in Section 5, this implementation is not generic because it uses Intergraph Geomedia Web Map. We plan on implementing the architecture using standards for the interfaces of the system and building modules independent of the supporting technology.
- *Develop tools to create geographic information systems based on the architecture.* Once the generic modules of the architecture are implemented, it is possible to build specific GIS applications by integrating the appropriate modules. It will be possible and desirable to define high-level languages and visual tools that allow a developer to easily select and integrate the modules for a specific GIS.

References

1. Laurini, R., Thompson, D.: Fundamentals of Spatial Informations Systems. The APIC Series Num. 37 - Academic Press (1992) ISBN: 0-12-438380-7.
2. Brisaboa, N.R., Cotelo Lema, J.A., Luaces, M.R., Viqueira, J.R.: State of the Art and Requirements in GIS. In: Proc. of the 3rd Mexican International Conference on Computer Science (ENC), Aguascalientes, Mexico (2001)
3. Burrough, P., McDonnell, R.: Principles of Geographical Information Systems. Oxford University Press (1998) ISBN: 0-19-823365-5.
4. Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Geographic Information Systems and Science. John Wiley & Sons (2001) ISBN: 0-471-49521-2.

5. Longley, P., Goodchild, M., Maguire, D., Rhind, D.: *Geographical Information Systems: Principles, Techniques, Management and Applications*. John Wiley & Sons (1999) ISBN: 0-471-32182-6.
6. Worboys, M.F.: *GIS: A Computing Perspective*. Taylor & Francis (1995) ISBN: 0-7484-0065-6.
7. Koubarakis, M., Sellis, T.K., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H., Scholl, M., Theodoulidis, B., N.Tryfona, eds.: *Spatio-Temporal Databases: The CHOROCHRONOS Approach*. Volume 2520 of *Lecture Notes in Computer Science*. Springer (2003)
8. Open GIS Consortium, Inc.: *OpenGIS Reference Model*. OpenGIS Project Document 03-040, Open GIS Consortium, Inc. (2003)
9. ISO/TC 211: *Geographic Information - Reference Model*. International Standard 19101, ISO/IEC (2002)
10. Open GIS Consortium, Inc.: *OpenGIS Geography Markup Language (GML) Implementation Specification, Version 3.00*. OpenGIS Project Document 02-023r4, Open GIS Consortium, Inc. (2003)
11. Open GIS Consortium, Inc.: *OpenGIS Styled Layer Descriptor Implementation Specification*. OpenGIS Project Document 02-070, Open GIS Consortium, Inc. (2002)
12. Open GIS Consortium, Inc.: *OpenGIS Web Feature Service Implementation Specification*. OpenGIS Project Document 02-058, Open GIS Consortium, Inc. (2002)
13. ISO/TC 211: *Geographic Information - Simple Feature Access - Part 2: SQL Option*. Draft International standard 19125-2, ISO/IEC (2000)
14. Open GIS Consortium, Inc.: *OpenGIS Simple Features Specification For OLE/COM*. Revision 1.1. OpenGIS Project Document 99-050, Open GIS Consortium, Inc. (1999)
15. Brisaboa, N.R., Coteló Lema, J.A., Fariña Martínez, A., Luaces, M.R., Viqueira, J.R.: *The E.I.E.L. Project: An Experience of GIS Development*. In: *Proc. of the 9th EC-GI & GIS Workshop (ECGIS)*, A Coruña, Spain (2003)

Intrusion Detection System for Securing Geographical Information System Web Servers

Jong Sou Park, Hong Tae Jin, and Dong Seong Kim

Computer Engineering Department, Hankuk Aviation University,
200-1 Hwajun-Dong Dukyang-Gu, Koyang-City, Kyonggi-Province, South Korea
{jspark,jhongtae,dskim}@hau.ac.kr

Abstract. Web servers which provide Geographical Information System (GIS) services are very vulnerable against attacks exploiting web-based programming errors. A traditional Intrusion Detection System (IDS), however, has limitations to detect web-based attacks because they usually use signature-based IDS. Therefore, we propose IDS based on Hidden Markov Model (HMM) for securing GIS web servers. We adopt HMM which has been achieved good performance in pattern recognition and intrusion detection. We demonstrate effectiveness and efficiency of our proposed system by carrying out several experiments.

1 Introduction

As Internet and Internet users are rapidly increasing and getting popularized in the world, the type of attacks are also changing. Traditional attack methods mainly targeted to exploit vulnerabilities contained in Operating System (OS) and network services, and also should build exploit codes with assembler such as buffer overflow attacks. These types of attacks, however, can hardly exploit known vulnerability because of firewall as well as constant patch of OS and countermeasure of network service management companies. But, in the case of web-based attack, administrator can not prevent their system from attacks by deploying existing firewall only because port 80 is legal traffic allowed for web service. In addition, Intrusion Detection System (IDS) evasion techniques using web-based attacks are too sophisticated, diversified and popularized so that general signature-based IDS can not detect web-based attacks [5,13]. Due to these reasons, attackers are changing their directions toward web-based attacks that take advantages of vulnerability in web services to penetrate into web server system. And because Geographical Information Service (GIS) system using web servers is a system of computer software, hardware and data in Internet environment, it is also very vulnerable in web-based attacks. GIS system contains very valuable information which can be used for transportation, aviation, etc. If the hacker can corrupt the information stored in GIS system, it could result in disastrous accidents. According to SecurityTracker Statistics and CERT/CC Incident and Vulnerability Trends, the execution of malicious code which exploit web server's programming errors is occupying the most portions of whole attacks [4,15]. Web-based attacks are very serious since attackers not only compromise

web server through Denial of Service (DoS) and/or sniffing but also acquire user and/or root privilege of web server. Most administrators use IDS usually to detect these kinds of web-based attacks, but most of IDS are signature based, it is thereby very difficult to detect most web-based attacks [2]. In this reason, web-based intrusion detection techniques to detect web-based attack have been proposed. Wen hui *et al.* [16] have proposed a novel two layer mechanism to detect intrusions against web-based database service but it is used for web-based database system only. And Ryutov *et al.* [17] have proposed integration of the generic authorization and access control API to provide dynamic intrusion detection and response for the apache web server. These approaches, however, focus on deploying access control and it only use simple application level intrusion detection system in which adopt signature-based intrusion detection model. And Vigna *et al.* [9] have presented a stateful intrusion detection system for www servers and that approach is centered around the state-transition analysis technique (STAT). This approach, however, can not deal with very sophisticated attacks that can not be expressed by states and transitions philosophy. Therefore, in this paper, we propose IDS using Hidden Markov Model (HMM) for securing GIS web server as well as conventional web server. We assume that web-based attacks using programming errors are sequences of states. And we also extract essential forms or types used in web-based attacks. HMM is an effective way to model states transition, and to detect unknown (hidden) states and have been shown to achieve good performance in pattern recognition [1,8]. We showed excellence of our proposed system's performance by carrying out several experiment.

This paper is organized as follows. In section 2, web-based IDS and HMM are presented. In section 3 and section 4, our proposed system and related experiments are introduced respectively. Finally, section 5 concludes our research work.

2 Related Works

2.1 Web-Based Intrusion Detection System

Intrusion Detection System(IDS) had been proposed first time by Anderson [11], and is divided into misuse detection model and anomaly detection model according to detection model, and into network based IDS and host based IDS according to location of audit data [2,7]. Misuse detection model is also called as signature-based model, and effectively detects well-known attack, but can not detect novel attacks. Anomaly detection method is used to improve this shortcoming. In the meantime, detection system was used into expenditure that watches whole state of general host. But, as web-based attack is diversified and sophisticated, IDS for web server was proposed to detect web attack. In early research, Wen hui *et al.* [16] have proposed a novel two layer mechanism to detect intrusions against web-based database service but it is only suitable for protecting web-based database system. In Ryutov *et al.*'s approach [17], they have presented the integration of the generic authorization and access control API to provide dynamic intrusion

detection and response for the apache web server. That approach, however, focus on deploying access control and it only use simple application level intrusion detection system in which adopt simple signature based intrusion detection model so that they cannot detect novel attacks. Vigna *et al.* [9] have presented a stateful intrusion detection system for www servers, and they adopted the state transition analysis technique (STAT). Using state transition is effective for modeling IDS, but that is not able to deal with very sophisticated attacks that cannot be expressed by simple states and transitions methods. On the other hand, HMM is more suitable for modeling web hacking since HMM does not require exact state transition activities. Therefore, we propose web server host-based anomaly detection model to mitigate these shortcomings and improve the performance of IDS using HMM.

2.2 Hidden Markov Model

HMM is finite state machine that has transition probability between each state, and state transition can not be observed directly, but each observation in a state is related to some probability distribution. HMM is a model which can get hidden information from observable information and composed of the following parameters [10,12].

- Hidden States: hidden states of the system
- Observable States: the observation from a state
- π Vector: vector representing initial probability of hidden states
- State Transition Matrix: transition matrix among hidden states, the transition probability from previous state to next state
- Confusion Matrix: probability that specific observation effect in a specific hidden state

HMM is expressed by $M = (\Pi, A, B)$ and each appears as following.

$$\Pi = (\pi_{ij}) : \text{probability in hidden states} \quad (1)$$

$$A = (a_{ij}) : \text{Transition probability, } Pr(x_{i_t} | x_{j_{t-1}}) \quad (2)$$

$$B = (b_{ij}) : \text{Confusion Matrix, } Pr(y_i | x_j) \quad (3)$$

Normal action modeling is the process to decide the parameters of HMM. We adjust M to maximize $Pr(O|M)$ probability which observation sequence O came from that model M . By this standard, we get the estimation value of each normal action sequence.

3 Proposed System

The structure of proposed system is depicted in Figure 1. If URL input enters, preprocessing log database extracts path and query distribution value used most frequently in log information through preprocessing algorithm. And using that information, we build optimized HMM model through model trainer and

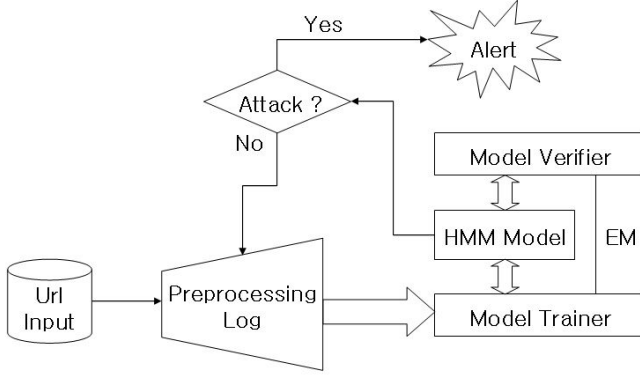


Fig. 1. Web-based IDS model using HMM.

verifier. Through preprocessing process, input URL data is divided into data for model trainer and model verifier. In the case of model trainer, it is used to learn the model through training module, and is selected trained HMM through model verifier. The parameter of HMM can be trained using EM (Expectation-Maximization) algorithm [6]. Therefore, we selected EM algorithm as a model verifier to get optimized model. If abnormal path or query which is not trained by learning algorithm enters, by applying some probability threshold, it generates alert signal.

3.1 URL Data

A basic preprocessing approach to analyze HTTP request of log have been proposed by C. Kruegel *et al.* [3,14]. We only focus on the GET requests. In addition, experiment data does not consider header data and POST/HEAD requests of GET requests. As the input, we used dataset which was extracted from successful GET requests which is valid HTTP code (200 status codes). Query strings are used to return parameter from referred resources, and these are divided by character “?”.

That is, the query is expressed as sorted list of parameters that have agreed values.

$$q = (a_1, v_1), (a_1, v_1), (a_2, v_2), \dots, (a_n, v_n), a_i \in A \text{ and } v_i \text{ is string} \quad (4)$$

The set S_q is defined by the subset $\{a_i, \dots, a_k\}$ of query q 's attributes. Figure 2 is an example of web log entry. In this case, S_q is $S_q = \{a_1, a_2\}$ and U_i is $U_i = \{Path, S_q\}$. Figure 2 shows an example of web log.

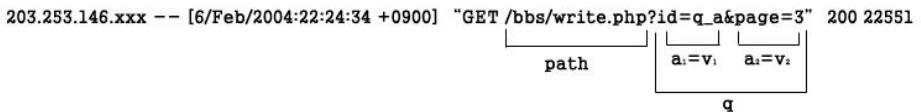


Fig. 2. An example of web log entry.

3.2 Log Preprocessing

It is impossible to extract all information because actual web services contain so many requests including path and query. Therefore, files, for instances, graphic file(gif, jpg, etc), flash file(swf), Cascade Style Sheet file(css), Java script file(js), and etc., in which do not include real information are removed from URL and we only focus on active document(asp, php, jsp, etc) files including actual information. Preprocessing algorithm is described in Table 1.

Table 1. Preprocessing Algorithm.

1. Extract all logs that have successful GET request from given web log.
2. Extract Path and all Query using Active documents Characteristics from the logs built in step 1.
3. Check the distribution threshold.
If (Path and Query Information > Threshold)
 Arrange Path directory and Query in alphabetical order
Else
 Remove it!
End
4. End if all log data are examined

3.3 Training and Test

Each data is divided by Training set (normal data only) and Test set(both normal and intrusion data). Training data set is to learn the parameters of normal model. Since web-based attacks using programming errors are usually manipulating path and query information as shown in Table 2, we focus on normal connection with path and query information to create HMM observation. As shown in Table 2, attack attempts are including distinguishing path and/or query informations such as “%3”, “cmd”, and “../”.

Table 2. Path and Query Information on attack types(examples).

Attack Type	Path	Query
Remote File Include	/Hakbu/data_18.jsp	dir, cmd
Directory Listing	/%3f/index.jsp	No query
Password	/.../.../.../.../etc/passwd	No query

4 Experiments

4.1 Experimental Dataset

As well-known web site log is hard to get due to personal private problem and web site management policy, we used the web log of some university for exper-

Table 3. Dataset Properties.

Field	Values
Time Interval	12 Days
Size(Mbytes)	600 Mbytes
URL Number	6,500,000
HTTP Queries	10,000
Path	30,000
IP(user) Number	500

iments. Even though we did not use practical web GIS information, we believe that web application services are very similar between ordinary web server and web server for GIS system, since GIS system also use commercial web server service programs and related application programs. We used 600Mbytes log data to preprocess 6,500,000 URL. We display the important information for the dataset as shown in Table 3.

Table 3 shows log file size and time interval while recording data. And it shows total number of HTTP query and path concerning that. Data used in an experiment for training and test is number of IP user who used web service in normal connection.

4.2 Log Preprocessing

After passing through preprocessing algorithm, log data is sorted as shown in Table 4. We set the threshold hold value τ (number of the Active Documents) to 30 as training set, and we make, for example, following observation sequence as classifying user's behavior and reading each path and query information to input URL. (S : Start(Connection established), E : End(disconnection), number represents the value sorted by path directory's alphabetical order through the algorithm.)

$$O = (o_1, o_2, \dots, o_T) = S \ P1bQ0 \ P9dQ0 \ P11aQ11abcg \ P9eQ0 \\ P9aQ0 \ P9fQ0 \ P9dQ0 \ P11aQ11abcdfg \ P9gQ0 \ P3bQ0 \ E \quad (5)$$

State numbers of observation sequence include start state(S) and end state(E). With these states, we build initial HMM model such as in Figure 3. Path and query that offer in web service are so wide that we can't use the whole information in learning algorithm. Therefore, in this paper, we choose data status number using the most used path and query frequency, and adjust data number to threshold value τ . Because there are many paths within path directory and query's combination, we considered these to hidden states.

4.3 Experimental Results

Effective extraction of users' pattern is limited, since the patterns of them change over time. Thus we constructed experiments to evaluate the model's performance

Table 4. Preprocessed Data Information(partially shown).

Number	Path Directory(P)	Active Document	Query(Q)
1	/	HaksMonBillWin.jsp(a) Index.jsp(b) notice.htm(c)	Hakbun(a), no(b) No query(Only Path exist) No query(Only Path exist)
2	/Academic/	0820.htm(a) notice.htm(c)	No query(Only Path exist) No query(Only Path exist)
3	/Busok/	data1.jsp(a) data2.jsp(b)	No query(Only Path exist) No query(Only Path exist)
4	/English/institutes/	index.jsp(a)	No query(Only Path exist)
5	/graduate/	ilban.htm(a)	No query(Only Path exist)
6	/Hakbu/	data1_4.jsp(a) data1_1professor.jsp(b)	No query(Only Path exist) No query(Only Path exist)
7	/ibhak/	data1_1_1_1.asp(a)	No query(Only Path exist)
8	/Intro/	data2_2.jsp(b) data2.jsp(b)	No query(Only Path exist) No query(Only Path exist)
9	/Living/	data8.jsp(a) data9.jsp(b) data10.jsp(c) data11.jsp(d) data12.jsp(e) data13.jsp(f) data14.jsp(g) data15.jsp(h) data16.jsp(i) data2_1.jsp(j) data2_2.jsp(k) notice.htm(l)	No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist) No query(Only Path exist)
10	/Recruit/	data1.jsp(a) data2.jsp(b)	No query(Only Path exist) No query(Only Path exist)
11	/servlet/	JMBorad.jsp(a)	Tablename(a), mode(b), boardpage(c), search-word(d), searchscope(e), category(f), no(g)
12	/UniInfo/	SearchStaff.jsp(a)	dept1(a), dept2(b),searchscope(e), category(f), no(g)

over a limited range of observed sequences in the constrained range of time. For the experiment, we have collected audit data from 500 users who have conducted several transactions as navigating web pages. We used 300 users for training parameter of HMM and 200 users for testing to build the parameters of HMM. After we built the HMM, we performed some experiments to evaluate the model on the detection rates and false positive (FP) rates with changing number of model states and a preprocessing threshold value. Through these experiments we know that the results are useful to detect intrusions when we have modeled normal behaviors by HMM.

Experiments focused on the number of the model states. We performed experiments on the detection rate observing normal query and path sequence against Remote File Attack, Password File Listing Attack and Directory Listing

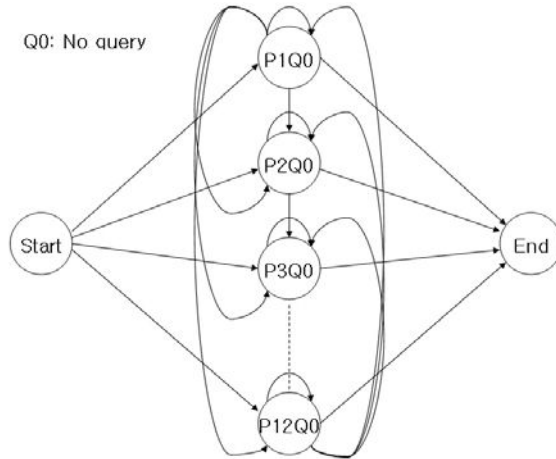


Fig. 3. Initial model using Path directory and Query.

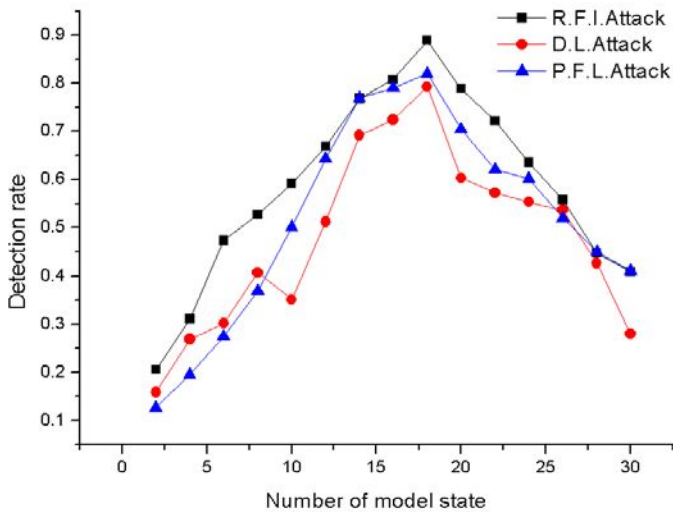


Fig. 4. Detection Rates vs. Number of Model States.

Attack. Figure 4 shows the result. By evaluating the number of states, our system is able to judge which a web log is attacks or not. We get the best detection rates when the number of states is 18 for three different attacks. This number of states was consistent and we believe that this consistency is a good factor.

Experiments focused on the threshold values. Figure 5 shows the False Positive (FP) rate with respected to threshold value. The FP rate gets very high value on low threshold value in Figure 5. If threshold value is low, it has small

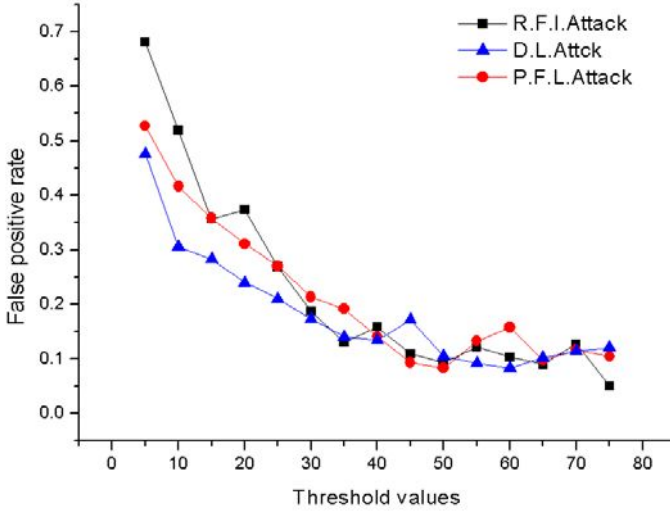


Fig. 5. False Positive rate vs. Threshold Values.

amount of information to learn normal data. Therefore, it decides normal data as intrusion and represents a high FP rate in the beginning.

5 Conclusions and Future Works

This paper proposed IDS for securing GIS web servers. Web-based GIS system is open for hackers to be penetrated into port 80 if no dedicated security tools exists. We need sanitization process for this critical infrastructure. If the hacker can corrupt the information stored in GIS system, by hacking into web server which is connected to GIS, it could result in disastrous accidents. Although HMM has already been adopted in IDS, from the best of our knowledge, we are proposing HMM for web server IDS for the first time. We experienced validity and efficiency of proposed system by performing several experiments on web log. We extracted normal connection information in web server's log, and through the preprocessing algorithm, we used path and query information as audit data, and we performed experiments on testing based on HMM. We are aware that this system shows effective way of detecting attacks. Future work includes applying more query instance and string value to this model, and we wish to increase attack detection range and efficiency. We also would like to reduce path and query loss information to get low FP rate. Also, more research works remains on getting efficient preprocessing algorithm to increase the performance of HMM. We used a simple preprocessing algorithm to check the validity of HMM. We also need to check the detection rate based on the seriousness of hacking. In other words, we need to be more concerned about web hacking which tries to modify information stored in GIS system.

Acknowledgements

This work was supported (in part) by the Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

References

1. Aggelos Pikrakis. *et al.*: Recognition of Isolated Musical Patterns Using Hidden Markov Models. Volume. 2445/2002. Title: Music and Artificial Intelligence: Second International Conference. (2002)
2. Aurobindo Sundaram.: An Introduction to Intrusion Detection. ACM crossroad Issue 2.4 April 1996 - Computer Security. (1996)
3. C. Kruegel, G. Vigna.: Anomaly Detection of Web-based Attacks, Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS '03) Washington. (2003)
4. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. *Int. J. Digit. Libr.* **1** (1997) 108–121
5. Craig H. Rowland.: Convert Channels in the TCP/IP Protocol Suite. <http://www.securitymap.net>
6. Dempster, A.P. *et al.*: Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of Royal Statistic Society B*, Vol. 39, pp. 1-38, (1977)
7. Dorothy E. Denning.: An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, 13(2):222-232, February (1987)
8. Fielding, K.H. Ruck, D.W.: Spatio-temporal pattern recognition using hidden Markov models. *Aerospace and Electronic Systems*, *IEEE Transactions on* pp: 1292-1300. (1995)
9. Giovanni Vigna. *et al.*: A Stateful Intrusion Detection System for World Wide Web Servers. 19th Annual Computer Security Applications Conference, (2003)
10. L.R. Rabiner and B.H. Juang.: An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.
11. James P. Anderson.: Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co., Fort Washington, PA, April (1980)
12. L. R. Rabiner.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), (1989)
13. Project Loki, daemon9/Alhambra.: Phrack magazine Volume Seven Issue Forty-Nine File 06 of 16. available at. <http://www.phrack.com> (1996)
14. R. Fielding *et al.*: Hypertext Transfer Protocol - HTTP/1.1. RFC 2616, June (1999)
15. SecurityTracker Statistics April 2001 - March 2002. available at. <http://www.securitytracker.com>
16. Shu Wenhui, Tan T H. Daniel.: A Novel Intrusion Detection System Model for Securing Web-based Database Systems. COMPSAC'01, (2001)
17. Tatyana Ryutov. *et al.*: Integrated Access Control and Intrusion Detection for Web Servers. *IEEE Transaction on Parallel and Distributed Systems*, (2003)

In-Route Skyline Querying for Location-Based Services

Xuegang Huang and Christian S. Jensen

Department of Computer Science, Aalborg University, Denmark
{xghuang, csj}@cs.aau.dk

Abstract. With the emergence of an infrastructure for location-aware mobile services, the processing of advanced, location-based queries that are expected to underlie such services is gaining in relevance. While much work has assumed that users move in Euclidean space, this paper assumes that movement is constrained to a road network and that points of interest can be reached via the network. More specifically, the paper assumes that the queries are issued by users moving along routes towards destinations. The paper defines in-route nearest-neighbor skyline queries in this setting and considers their efficient computation. The queries take into account several spatial preferences, and they intuitively return a set of most interesting results for each result returned by the corresponding non-skyline queries. The paper also covers a performance study of the proposed techniques based on real point-of-interest and road network data.

1 Introduction

Location-based services (LBSs) utilize consumer electronics, mobile communications, positioning technology, and traditional map information to provide mobile users with new kinds of on-line services. Examples include location-sensitive information services that identify points of interest that are in some sense nearest and of interest to their users and that offer travel directions to their users. Data management is a core aspect of the provisioning of LBSs, and advanced services pose new challenges to data modeling as well as update and query processing.

Using the moving users' freedom of movement, three scenarios for LBSs can be distinguished. First, unconstrained movement is characterized by the mobile users being able to move freely in physical space. Next, constrained movement occurs when movement is constrained by obstacles, e.g., buildings and restricted areas. The third scenario is that of network-constrained movement, which is this paper's focus. Here, user movement is restricted to a transportation network, and Euclidean distances are generally of little use. Rather, the notion of travel distance takes center stage, and query processing techniques must use this distance notion.

This paper considers so-called in-route queries. These assume that the user's destination is known, in addition to the user's current location; and they assume that an anticipated route towards the destination is known. This setting is motivated by the observation that few mobile users move about aimlessly, but rather travel towards a known destination along a known route. Such routes can be obtained from navigation systems or past behavior [3].

Next, users are likely to take several spatially-related criteria, with varying weights, into account when deciding on points of interest to visit. As examples, a user looking

for a gas station may prefer to minimize detour rather than distance, while a user searching for an emergency room is likely to be interested in minimizing the distance and is insensitive to the detour. We propose to use the mechanism inherent in the skyline operator [2] to balance several criteria, and we generalize the skyline mechanism to enable queries that return larger result sets from which the user can then choose. The skyline mechanism returns a result if no other result exists that is better with respect to all the criteria considered. This is useful when a total ordering cannot be defined on the space of all criteria.

The paper offers algorithms for in-route k th order skyline queries, and it covers performance studies with real point-of-interest and road network data. Although focus is on spatial preferences, non-spatial preferences can be integrated into the contribution.

Inspired by recent work by Speiçys et al. [17] and Hage et al. [8], we use generic data structures for representing a road network and points of interest within the network. These structures separate the network topology from the points of interest, which is important for maintainability, and are kept simple, so that the paper's contributions are broadly applicable.

Query processing for network-constrained moving objects has recently received attention in the research literature. Data models and query processing frameworks [8, 15, 17] as well as indexing methods [6, 13] have been proposed for this scenario. Nearest neighbor querying has attracted the most attention [9, 18, 20]. The paper builds on these advances, and it considers in detail the approaches for in-route nearest neighbor query proposed by Shekhar and Yoo [20].

Next, the skyline operator was recently introduced into the context of databases [2]. Several skyline algorithms have since appeared [4, 11, 12, 14, 21]. Most algorithms assume that the points to be queried are stored in an R-tree like structure [7]. This paper generalizes the skyline operator and applies it in a new setting; and it does not assume the presence of an index, but rather uses different means of pruning the search space.

Three contributions may be identified. First, the paper advances the idea of in-route movement with associated spatial preferences and consequent k th order skyline queries. Second, techniques for computing such queries for different kinds of in-route movement are presented. Third, a performance evaluation using real point-of-interest and road network data is covered.

Section 2 introduces the data model, the road network representation, and basic algorithms. The next section discusses the movement of in-route users and the corresponding nearest-neighbor based preferences used in queries. Then Section 4 proposes the query algorithms, and Section 5 covers the performance experiments. The last section summarizes and offers research directions.

2 Data Model and Basic Algorithms

2.1 Problem Statement

As discussed, we assume that the users of LBSs are road-network constrained; for example, users may drive by car or may be bus passengers. Next, a number of facilities or so-called points of interest, e.g., gas stations and supermarkets, are located within the road network. We assume that users query the points of interest en route towards

a destination. Specifically, the users issue queries with the purpose of finding a point of interest to visit while moving along a pre-defined route towards a given destination. Having found a point of interest, the user visits this point and then continues towards the destination. Section 3 discusses distance-related preferences.

We term users *query points* and the points of interest *data points*. We proceed to model the problem scenario, including the road network, data points, and the current location, route, and destination associated with query points.

2.2 Data Model

A *road network* is a labeled graph $RN = (V, E)$, where V is a set of vertices and E is a set of edges. A vertex models an intersection or the start or end of a road. An edge e models the road in-between two vertices and is a three-tuple $e = (u, v, w)$, where $u, v \in V$ are vertices and w is the length (or, weight) of e . We assume that an edge may be traversed in both of its directions.

Next, a *data point* dp is a two-tuple $dp = (e, pos_u)$, where e is the edge on which dp is located and pos_u represents the distances from vertex u of e to dp . The distance from v of e to dp is then $w - pos_u$. Note that adding and removing data points does not affect the graph itself, which is important for maintainability in practice. A data point found by an algorithm proposed in this paper is a *candidate point*, $cp = (dp, x_1, \dots, x_m)$, where the x_i are attributes generated by the particular algorithm. A *query point* qp has the same format as a data point, and pos_u denotes the current distance from vertex u of e to the (moving) point.

Finally, a *route* is given by a sequence of neighboring vertices $\langle r_0, r_1, \dots, r_l \rangle$, where $r_i \in V, i = 0, \dots, l$. We assume the current location c of the query point is on the edge between r_0 and r_1 , as edges of a route that have already been traversed are of little interest in our context. It is also assumed that the query point cannot make a “u-turn” between its current location c and r_1 . The destination associated with a route is the last vertex, r_l . A desired destination that is not represented by a vertex may be handled by introduction of a temporary vertex into the road network, or by running the algorithms twice, once

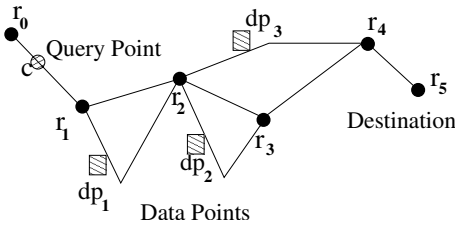


Fig. 1. Data Model Concepts.

for each neighbor vertex of a destination on an edge, followed by refinement of the results. Figure 1 illustrates the concepts defined above.

2.3 Disk-Based Road Network Representation

A road network is stored as two paginated adjacency lists: one for vertices and one for edges. To obtain locality in operations, vertices are organized in pages according to their Hilbert values.

Each element in the adjacency list of a vertex v corresponds to an adjacent vertex and contains 3 entries: a pointer to the page that contains the adjacent vertex, a pointer to the page in the adjacency list of edges that contains the corresponding edge, and the length of this edge.

The adjacency list for edges records the relations between edges and data points. The adjacency lists for two edges e_1 and e_2 are put in the same page if their vertices are in the same page. Each element in an edge's adjacency list contains information about a data point located on this edge.

2.4 Basic Operations and Algorithms

Skyline Operation. The skyline operation retrieves those points in an argument data set that are not dominated by any other points in the set. More specifically, consider a set of points in l -dimensional space. Point p_1 dominates point p_2 if p_1 is at least as good as p_2 in all dimensions and better than p_2 in at least one dimension [2]. Here, we define “better” as “smaller than.” For example, assume we have points $p_1 = (2, 4)$, $p_2 = (3, 5)$, $p_3 = (1, 6)$ in two-dimensional space. Point p_1 dominates p_2 as it is better than p_2 in all dimensions. But p_1 and p_3 do not dominate each other. So p_1 and p_3 are the skyline points in set $\{p_1, p_2, p_3\}$.

We are interested in finding not only those points that are not dominated by any other point, but also the points that are dominated by less than k other points, termed the k th order skyline points. This generalization is similar in spirit to the generalization of the nearest neighbor query to the k nearest neighbors query.

Many existing skyline algorithms assume that the argument points are pre-processed into a data structure such as the R-tree. In contrast, we aim to prune unnecessary search of candidate points before the skyline operation, and we assume instead that the candidate points are organized in a simple main-memory list structure.

For a point set $P = \{p_1, \dots, p_n\}$ and $p_i = (dp, x_1, \dots, x_m)$, we denote the k th order skyline operation by $SKYLINE(k, P, \{d_1, \dots, d_l\})$, where $\{d_1, \dots, d_l\}$ are the dimensions to be considered. To compute the result, each point is compared to all other points to determine its skyline order. The set of points with skyline order less than k is the result.

Distance Function. We define the distance between two vertices v_i, v_j , denoted by $D(v_i, v_j)$ as the sum of lengths of the edges along the shortest path between these vertices. We use Dijkstra's single-source shortest path algorithm [5] for computing such distances. Since edges may be traversed in both directions, $D(v_i, v_j) = D(v_j, v_i)$.

In addition, for two vertices v_i, v_j along a route R , the road distance from v_i to v_j along the route is denoted by $\mathbf{D}(v_i, v_j, R)$ and is defined as the sum of edge lengths between each two neighboring vertices in the vertex sequence of route R . If v_i is after v_j along the route, the edge lengths are counted as negative so that $\mathbf{D}(v_i, v_j, R) < 0$. Note that $D(v_i, v_j) \leq |\mathbf{D}(v_i, v_j, R)|$. We also use $\mathbf{D}(c, r, R)$ to denote the distance from the query point's current location c to a vertex r along route R . This distance is the sum of the distance from c to r_1 and $\mathbf{D}(r_1, r, R)$.

Spatial Range and Nearest Neighbor Queries. We use $RQ(v, d)$, where v is a vertex and d is a real-valued range, to denote the range query that returns all data points that are within distance d of vertex v . We use a traditional best-first search algorithm [10] extended by the reading of data points from edges for computing this query.

We use $NNQ(k, v)$, where k is a positive integer and v is a vertex, for denoting the set of (up to) k data points that satisfy the condition that no other data points are nearer to vertex v than any point in the argument data set. A number of algorithms exist that compute this query [9, 15, 18, 20], and we use an algorithm that is based on ideas from these. Briefly, it is a traditional best-first search, extended incrementally to read data points from edges and to retrieve the k nearest ones.

We also use an algorithm that, for a vertex v , finds the (up to) k nearest neighbor data points that are also within distance d of v . We denote this query by $RNNQ(k, v, d)$. This algorithm combines the range and nearest neighbor algorithms.

3 Classification of In-Route Nearest Neighbor Queries

We first discuss the possible distance-related preferences for in-route skyline queries. Then we propose a query classification based on the movement of the query point.

3.1 Distance-Related Preferences

Two distance-related preferences are of particular interest in relation to a query point, a data point, and a destination.

Total distance difference: The pre-defined movement is from the current location to the destination along the route. To visit a data point, the query point will change its movement, to go from the current location to the data point and then to the destination. The total distance difference is the larger or smaller distance, compared to that of the pre-defined movement, that the query point must travel to visit the data point and then go to the destination.

Distance to the data point: The distance to the data point is the distance the query point needs to travel to reach the data point.

Optimizing for one preference may adversely affect the other. So if short total distance *as well as* short distance to the data point are of interest, combining these in a skyline query is natural. For brevity, we will in the sequel denote the total distance difference as the “detour,” while the distance to the data point is denoted as the “distance.”

To find the skyline points, we need to first associate distance and detour values with the data points. Our focus will be on the process of searching for candidate points in the skyline operation. We proceed to classify the possible movements of a query point and consider the distance and detour values for each classification. The resulting classification may be used when pruning the search.

3.2 Classification

Although a query point may move unpredictably, it will always be leaving for a data point at a vertex along its route, and its return to the route can also be characterized by

a vertex along the route. We consider next the general case and then two special cases that are likely to be of interest in specific real-world uses. Recall that without loss of generality, the query point's current location c is between r_0 and r_1 along the route and the destination r_l is the last route vertex. All three cases are shown in Figure 2.

General Case. We first discuss the general case of a query point's movement. A query point issues a query en route towards its destination. The user selects a data point from the result, visits that point, and then proceeds to the destination.

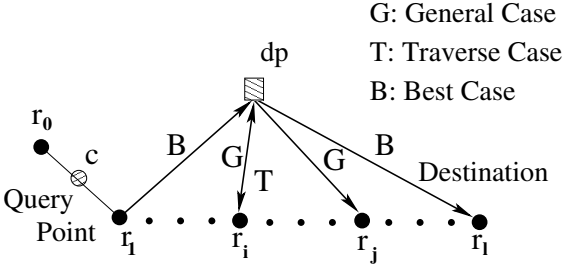


Fig. 2. Classification.

vertices in the pre-defined route and r_j may possibly be the destination. In Figure 2, if the route is R , the distance from the query point's current location to data point dp is $D(c, r_i, R) + D(r_i, dp)$, and the detour is $D(r_i, dp) + D(dp, r_j) - D(r_i, r_j, R)$.

Traverse Case. In the special case considered here (see label “T” in Figure 2), the query point leaves the route at r_i for data point dp and returns to the route at r_i . In this case, the “leaving” and “returning” vertices are the same vertex. The distance to the data point is the same as the general case, while the detour value is $D(r_i, dp) + D(dp, r_i) = 2D(r_i, dp)$ so that unless the data point is on the route, the detour is bigger than zero. This case applies to users who are faithful to their route, which may be the case for, e.g., tourists following a scenic route.

Best Case. In this case (see label “B” in Figure 2), the query point leaves the route for data point dp at r_1 and then goes directly towards the destination in the route. The distance and detour for this case are obtained by replacing i, j by 1 and l in the formulas given for the general case. Observe that in this case, the route carries little significance to the user.

In addition to the traverse and best cases, other special types of movement are possible. We simply categorize these as belonging to the general case. So in the next section, we provide algorithms for the traverse case and the best case; and based on the observations from these algorithms, we present an algorithm for the general case of the in-route nearest neighbor skyline query.

4 Algorithms for In-Route Skyline Queries

One basic approach to compute the *SKYLINE* query is to scan all data points, determining for each point whether it is a skyline point. Assuming a large number of data points, this is too costly. A strategy should be employed that prunes unnecessary search.

4.1 Algorithm for the Traverse Case

In the traverse case, the query point leaves the route and returns at the same vertex. Shekhar and Yoo [20] have previously considered this case, and our algorithm for this case is similar to theirs, the main differences being that we use road distance and any k , while they use Euclidean distance and assume $k = 1$.

Given an order k , a query point's current location c , and a route $R = \langle r_0, r_1, \dots, r_l \rangle$, the algorithm for the traverse case, *TraverseSQ*, is seen below. Two auxiliary queues, T and P , are used to store the result data points of nearest neighbor queries and the candidate points for the skyline query.

```

(1) procedure TraverseSQ( $k, c, R$ )
(2)    $P \leftarrow \emptyset$ 
(3)    $T \leftarrow NNQ(k, r_1)$ 
(4)   for each  $t \in T$ 
(5)      $dis \leftarrow \mathbf{D}(c, r_1, R) + D(r_1, t)$ 
(6)      $det \leftarrow 2D(r_1, t)$ 
(7)      $P \leftarrow P \cup \{(t, dis, det)\}$ 
(8)    $d \leftarrow$  distance from  $r_1$  to its  $k$ th neighbor
(9)   for each  $r_i, i = 2, \dots, l$ 
(10)     $T \leftarrow RNNQ(k, r_i, d)$ 
(11)    if  $T$  not empty
(12)      for each  $t \in T$ 
(13)         $dis \leftarrow \mathbf{D}(c, r_i, R) + D(r_i, t)$ 
(14)         $det \leftarrow 2D(r_i, t)$ 
(15)         $P \leftarrow P \cup \{(t, dis, det)\}$ 
(16)       $d_1 \leftarrow$  distance from  $r_i$  to its  $k$ th neighbor
(17)      if  $d_1 < d$ 
(18)         $d \leftarrow d_1$ 
(19)   return (SKYLINE( $k, P, \{dis, det\}$ ))

```

The algorithm is based on the following observations (Figure 3). Assume r_1 and r_2 are route vertices. Let data point dp_1 be the k th nearest neighbor of r_1 and let data point dp_2 be the $k + 1$ st nearest neighbor of r_1 . Then $D(r_1, dp_2) > D(r_1, dp_1)$. Since the query point's current location c is before r_1 , it is obvious that the distance from the query point's current location to dp_1 , via r_1 , is smaller than the distance to dp_2 . Also in this case, the detour of dp_1 is $2D(r_1, dp_1)$ and the detour of dp_2 is $2D(r_1, dp_2)$. So dp_1 dominates dp_2 . Since dp_1 is the k th nearest neighbor of r_1 , dp_2 is at most in the $k + 1$ st order skyline. As we perform nearest neighbor search incrementally, there is no need to continue after the k th nearest neighbor of r_1 is found.

Let $d_1 = D(r_1, dp_1)$. A k range nearest neighbor query is issued at r_2 with range d_1 (dashed polygon in Figure 3). If the k th nearest neighbor to r_2 , e.g., dp_3 is found, it will be collected as a candidate point. And its distance to r_2 , i.e., $d_2 = D(r_2, dp_3)$, will be compared to d_1 to possibly obtain a smaller search range at the next route vertex.

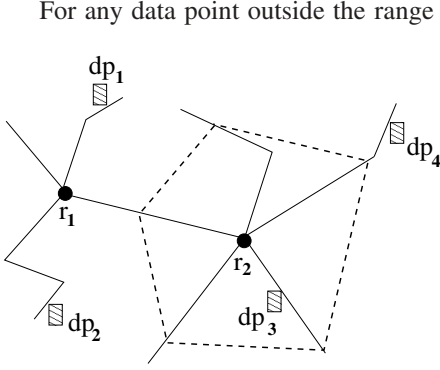


Fig. 3. Observations for the Traverse Case.

For any data point outside the range of d_1 , e.g., dp_4 , it is obvious that the detour of dp_4 is bigger than the detour of dp_1 . The distance from the user's current location to dp_1 is $D(c, r_1, R) + d_1$, and the distance to dp_4 is $D(c, r_1, R) + D(r_1, r_2, R) + D(r_2, dp_4)$. So any data point like dp_4 will be dominated by dp_1 , which means that such data points will at most be in the $k + 1$ st order skyline. So there is no need to expand the search beyond d_1 at r_2 .

Note that if the query point can leave for the data point at only a few specified vertices along the route, algorithm *TraverseSQ* can easily be adapted to handle this special case more efficiently.

The complexity of the algorithm is dominated by the two basic operations, *NNQ* in line 3 and *RNNQ* in line 10, the inner loop in lines 12–15, and the skyline operation. The two primitive operations can be seen as network expansion processes that use Dijkstra's single source shortest path algorithm to search for data points along the edges. Let $|E|$ be the number of edges and $|V|$ the number of vertices in the road network. The complexity of this operation is then $O(|E| + |V|\log|V|)$, if an F-heap is employed [5]. Taking into account that (up to) k data points are to be retrieved, each of the two operations has a complexity of $O(|E| + |V|\log|V| + k)$. We assume that the data points on an edge can be accessed in the order of their distances from the beginning (and ending) vertex of the edge. The inner loop runs at most k times at each route vertex. The complexity is $O(k|R|)$, where $|R|$ is the number of route vertices. The skyline operation, as described in Section 2.4, compares each argument point with all other argument points to determine its skyline order. Since up to k data points may be found at each route vertex, the complexity of the skyline operation is then $O(k^2|R|^2)$. Thus, the complexity of the *TraverseSQ* algorithm can be given as $O(|R|(|E| + |V|\log|V| + k^2|R|))$. Section 5 provides a much more detailed empirical study of the performance of the algorithm.

4.2 Algorithm for the Best Case

The skyline algorithm for the best case only takes the current location c , vertex r_1 and destination r_l on the route into consideration. For any skyline point dp found by this algorithm, the movement of the query point is from c to dp via vertex r_1 and then from dp to r_l .

The algorithm is based on observations illustrated in Figure 4 and explained next. Let r_i and r_j be vertices on route R and let the query point's movement be from r_i to the data point and then to r_j . Then, to find all the candidate points, two range queries are issued at r_i and r_j . Let dp_1 be the k th nearest neighbor of r_i and let $d_1 = 2D(r_i, dp_1) +$

$D(r_i, r_j, R)$ and $d_2 = D(r_i, r_j, R) + D(r_i, dp_1)$. Then d_1 is the distance from r_i to the k th neighbor data point dp_1 and the back, plus the distance from r_i to r_j . So, that all the k nearest neighbor data points from route vertex r_i should be found within a range around r_i of size d_1 . Next, d_2 is the distance from dp_1 to r_i plus the distance from r_i to r_j along the route. Consideration of a range of size d_2 around vertex r_j is sufficient to find all the k nearest neighbor data points of r_i . We proceed to discuss this observation in detail.

With d_1 and d_2 , the two range queries are then $RQ(r_i, d_1, RN)$ (dotted polygon in Figure 4) and $RQ(r_j, d_2, RN)$ (dashed polygon in Figure 4). Let all the data points in the road network be in set S , let the data points found by the range query using r_i be in set S_i , and let the data points found by range query using r_j be in set S_j .

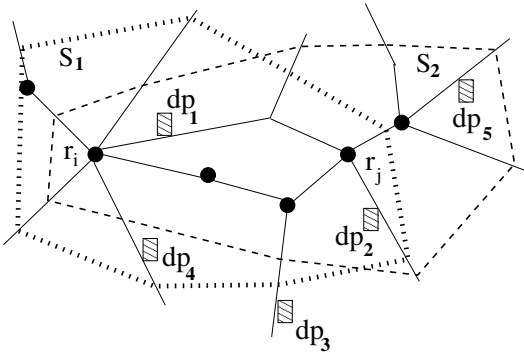


Fig. 4. Observations for the Best Case.

generality, we compare dp_1 with dp_3 , dp_4 , and dp_5 from Figure 4. It can be observed that dp_3 is inside neither S_i nor S_j . So $D(r_i, dp_3) > D(r_i, dp_1)$ and $D(dp_3, r_j) > D(dp_1, r_j)$. So the distance from r_i to dp_1 is smaller than that to dp_3 . Also, since the whole road length from r_i via dp_1 to r_j is smaller than from r_i via dp_3 to r_j , the detour of dp_1 is smaller than that of dp_3 . So any data point such as dp_3 will be dominated by dp_1 .

For dp_4 , since dp_1 is the nearest neighbor to r_i , $D(r_i, dp_4) \geq D(r_i, dp_1)$, and as dp_4 does not belong to S_j , $D(dp_4, r_j) > D(dp_1, r_j)$. So dp_4 is also dominated by dp_1 .

For dp_5 , we have $D(r_i, dp_5) > 2D(r_i, dp_1) + D(r_i, r_j, R)$, so the distance from r_i to dp_5 is bigger than the distance to dp_1 . And the detour from r_i to r_j via dp_5 is $D(r_i, dp_5) + D(dp_5, r_j) - D(r_i, r_j, R) > 2D(r_i, dp_1)$. As the detour of dp_1 is less than or equal to $2D(r_i, dp_1)$, dp_5 is dominated by dp_1 .

Since dp_1 is the k th nearest neighbor of r_i , all the other $k - 1$ st nearest neighbors whose distances to r_i are smaller than that of dp_1 are also in $S_i \cap S_j$. So there are at least k data points in $S_i \cap S_j$. And based on the discussion above, data points not in $S_i \cap S_j$ are dominated by the points inside this region, making them at most $k + 1$ st order skyline points. Thus, there is no need to check data points not in $S_i \cap S_j$.

The algorithm that implements the above-described search procedure is given next. It takes an order k , a query point's current location c , two route vertices r_i and r_j , and

It is clear that dp_1 belongs to $S_i \cap S_j$: $dp_1 \in S_i$ since $D(r_i, dp_1) \leq d_1$; and $D(dp_1, r_j) \leq D(dp_1, r_i) + D(r_i, r_j, R)$, so $dp_1 \in S_j$. Also the detour of dp_1 is less than or equal to $2D(r_i, dp_1)$ since the length of the shortest path from r_i via dp_1 to r_j is no longer than $2D(r_i, dp_1) + D(r_i, r_j, R)$. It should be noted that $dp_2 \in S_j$, but dp_2 may not be in $S_i \cap S_j$.

We proceed to explain why data points in $S_i \cap S_j$ dominate other data points. Without loss of

a route R as arguments. Two auxiliary queues, T_1 and T_2 , store the data points found in the two range queries at r_i and r_j . Queue P stores the result set of candidate points.

- (1) **procedure** $DNNQ(k, c, r_i, r_j, R)$
- (2) $T_1, T_2 \leftarrow \emptyset; P \leftarrow \emptyset$
- (3) $d \leftarrow$ distance from r_i to its k th neighbor
 \quad // compute by this by expansion from r_1 and pause when the k th neighbor is found
- (4) $d_1 \leftarrow 2d + \mathbf{D}(r_i, r_j, R)$
- (5) $d_2 \leftarrow \mathbf{D}(r_i, r_j, R) + d$
- (6) $T_1 \leftarrow RQ(r_i, d_1)$ // continue the paused expansion
- (7) $T_2 \leftarrow RQ(r_j, d_2)$
- (8) **for each** $t \in T_1 \cap T_2$
- (9) $dis \leftarrow \mathbf{D}(c, r_i, R) + D(r_i, t)$
- (10) $det \leftarrow D(r_i, t) + D(t, r_j) - \mathbf{D}(r_i, r_j, R)$
- (11) $P \leftarrow P \cup \{(t, dis, det)\}$
- (12) **return** P

With the input arguments k , c , and route $R = \langle r_0, r_1, \dots, r_l \rangle$, the skyline query algorithm for the best case, $BestSQ$, follows.

- (1) **procedure** $BestSQ(k, c, R)$
- (2) $P \leftarrow DNNQ(k, c, r_1, r_l, R)$
- (3) $P \leftarrow SKYLINE(k, P, \{dis, det\})$
- (4) **return** P

Algorithm $DNNQ$ suggests that to find candidate points for any type of movement, one needs to perform two range queries, at the “leaving” and “returning” vertices, and then collect data points in the intersection of the two ranges. We proceed to present a general skyline algorithm that is applicable to any type of movement of the query point.

The complexity of the $BestSQ$ algorithm is dominated by the two range queries in lines 6 and 7 and the iteration in lines 8–11 in algorithm $DNNQ$, as well as the skyline operation. The range queries can be treated as network expansion processes. In the worst case, the whole network and all data points DP are read, yielding a complexity of $O(|E| + |V|\log|V| + |DP|)$. The iteration has complexity at most $O(|DP|^2)$. So, taking also into account the complexity of the skyline operation, as described in the previous section, the complexity of algorithm $BestSQ$ is $O(|E| + |V|\log|V| + |DP|^2)$.

4.3 Algorithm for the General Case

While the two special cases may be prevalent, other cases exist. We thus provide an algorithm that works independently of the kind of movement of the user. The algorithm is based on the observation from $DNNQ$ that since there always needs to be range queries at all the route vertices, we can issue range queries at each route vertex with the biggest range once and for all and then check the data points in the intersections of each pair of ranges.

Two auxiliary structures are used in the algorithm. First, a set of queues $T = \{T_1, \dots, T_l\}$ store result data points found from the range queries at route vertices. To

retrieve the data points in T_i within a distance range d , an auxiliary function $Retr(T_i, d)$ is used. Next, a float array D stores the distance from each route vertex to its k th nearest neighbor.

The general in-route nearest neighbor skyline algorithm, *GeneralSQ*, uses the same arguments as does *TraverseSQ* and is given below.

```

(1) procedure GeneralSQ( $k, c, R$ )
(2)    $P \leftarrow \emptyset; T_i \leftarrow \emptyset; D_i \leftarrow 0, (i = 1, \dots, l)$ 
(3)   for each  $r_i \in R$ 
(4)      $d \leftarrow$  maximum range for  $r_i$ 
           // computed by comparing the range size for each pair of route vertices
(5)      $T_i \leftarrow RQ(r_i, d)$ 
(6)      $D_i \leftarrow$  distance from  $r_i$  to its  $k$ th neighbor
(7)     for each pair  $r_i, r_j \in \langle r_1, \dots, r_l \rangle, i \neq j$ 
(8)        $d_1 \leftarrow 2D_i + \mathbf{D}(r_i, r_j, R)$ 
(9)        $d_2 \leftarrow \mathbf{D}(r_i, r_j, R) + D_i$ 
(10)      for each  $t \in \{Retr(T_i, d_1) \cap Retr(T_j, d_2)\}$ 
(11)         $dis \leftarrow \mathbf{D}(c, r_i, R) + D(r_i, t)$ 
(12)         $det \leftarrow D(r_i, t) + D(t, r_j) - \mathbf{D}(r_i, r_j, R)$ 
(13)         $P \leftarrow P \cup \{(t, dis, det)\}$ 
(14)    $P \leftarrow SKYLINE(k, P, \{dis, det\})$ 
(15)   return  $P$ 

```

The algorithm first issues range queries at each route vertex with the biggest range to obtain all possible data points. Then, for each pair of route vertices r_i and r_j , the algorithm finds candidate data points assuming that the query point's movement is from r_i to the data point and then to r_j . The results are then filtered by the *SKYLINE* algorithm. Finally, all skyline points are collected. It is possible that one data point is collected more than once because of different kinds of movement.

Considering the algorithm's complexity, we observe that the *RQ* operation in line 5 is issued for each route vertex. This yields a complexity of $O(|R|(|E| + |V|\log|V| + |DP|))$, as discussed earlier. The iteration in line 7 executes $|R|(|R| - 1)/2$ times. At each iteration, the nested iteration starting in line 10 checks all data points found for a pair of route vertices, which is all data points in the worst case. This yields a complexity of the entire iteration of $O(|DP|^2|R|^2)$. The skyline operation has complexity $O(|DP|^2|R|^4)$. Thus, the complexity of algorithm *GeneralSQ* is $O(|R|(|E| + |V|\log|V|) + |R|^4|DP|^2)$.

It can be seen from the complexity analysis that if there are many route vertices, it will be relatively costly to gather the candidate points for each intersection of ranges, since there will be many pairs of vertices. We observe that *GeneralSQ* may work well if there are only a few possible "leaving" and "returning" vertices along the route. This corresponds to a scenario where a traveler would like to leave the route for a point of interest and return at some "familiar" or "well-known" locations along the route.

We proceed to offer a more detailed study of the performance of this and the previous two algorithms covered in this section.

5 Experimental Evaluation

The experiments described here use a real-world representation of the road network covering the municipality of Aalborg, Denmark. The road network contains 11,300 vertices, 13,375 bi-directional edges, and 279 data points that can be accessed via the network. So the data density, the number of data points over the number of edges of a road network, is 2%.

We define the size of a route as the number of vertices it contains. The page size is set to 4k bytes, and an LRU buffer is employed for simulation. A total of 136 pages contain adjacency lists of vertices, and 3 pages contain adjacency lists of edge. For the *GeneralSQ* algorithm, we use 10% of the number of route vertices as the possible “leaving” and “returning” vertices and assume that the “leaving” vertices are before the “returning” vertices.

To evaluate the effect of route size, skyline order (k), and buffer size, three experiments are conducted that measure query performance in terms of CPU time, I/O cost, and number of candidate points. The CPU-time is the actual running time for these algorithms. The I/O cost is the amount of pages read into the LRU buffer. The number of candidate points is the count of the candidate points that are found in these algorithms before the skyline operation. Since an in-memory skyline algorithm is used in the algorithms, the skyline computation is only evaluated in terms of CPU-time.

In the experiments, random routes are generated. The current location and destination are assumed to be the first and last vertices in the route. For each algorithm, we execute a workload of 100 queries and report the average performance. The experiments were performed on a Pentium IV 1.3 GHZ processor with 512 MB of main memory and running Windows 2000. The C++ programming language was used. To determine the variations in results due to external factors beyond our control (e.g., operation system tasks), we executed the same workloads multiple time. The results obtained exhibit only insignificant variations across repeated executions.

It should also be noted that the I/O costs for all the algorithms are measured using simulation. These costs are thus independent of the hardware and operating system used. Also, for the algorithms we have discussed, the I/O cost is more significant than the CPU time.

5.1 Experiment on the Effect of Route Length

In this experiment, the skyline order k is set to 5, and the buffer size is 10% of the road network size. The route size is varied from 50 to 300 to check the CPU time and I/O cost of the three algorithms. The results are shown in Figure 5.

It is clear that the *TraverseSQ* algorithm has the best performance and that *BestSQ* is in turn better than *GeneralSQ*. We proceed to discuss the findings for each algorithm.

It can be observed that the cost of *TraverseSQ* grows slightly with an increase in route size. Since the data density of the road network is 2%, when the number of route vertices is small, the data points are far from the route vertices. CPU and I/O costs grow slightly as the route size increases because a search process is required for each route vertex. But when the amount of route vertices grows, chances of finding a k -nearest neighbor close to a route vertex are higher, so that the search range for subsequent

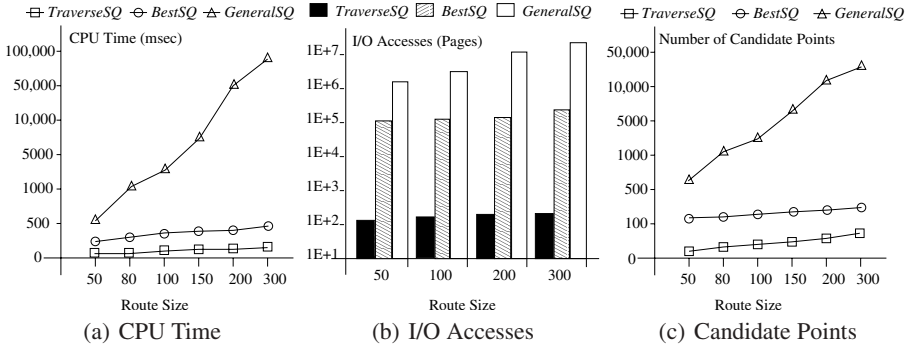


Fig. 5. Performance of Algorithms Versus Route Size.

vertices is smaller. Also, it can be seen from Figure 5(c) that the numbers of candidate points are so small that the skyline operation has no impact on the overall performance of *TraverseSQ*.

The overall cost of *BestSQ* grows slightly with an increase in route size. This is because the sizes of the two range queries in the *BestSQ* algorithm depend both on the route size and the distance from the first route vertex to its k th nearest neighbor data point. When the route size is small, the distance from the first route vertex to its k -nearest data point has the biggest effect on the cost. As the route size grows, its effect on the performance of the algorithm increases little by little. Note that in this case, the maximum number of candidate points can be found is 279. Because of this, the skyline operation has only minor impact on the overall performance.

It can be seen from Figure 5 that the cost of *GeneralSQ* increase drastically with the growth of route length. When the route size grows, more range queries are issued. The size of these range queries also grows since the route is longer. Also, since the number of candidate points exceeds 5,000 when the route size is 150, secondary-memory skyline processing is required.

5.2 Experiment on the Effect of k

In this experiment, the route size is set to 100, and the buffer is 10% of the size of the road network. The skyline order k is varied from 1 to 50. The results are shown in Figure 6. The general performances of the three algorithms follow the same trends as in the experiment on the effect of route size. However, when $k = 50$, *TraverseSQ* and *BestSQ* are quite close. This is because when $k = 50$, all the data points have been scanned by both algorithms. We proceed to discuss each individual algorithm.

It can be observed that the cost of the *TraverseSQ* algorithm increases with an increase in k . When k is increased from 10 to 50, the increase is quite significant. This is because the performance of *TraverseSQ* depends mostly on the k nearest neighbor search at the first several vertices in the route, so that when the k is increased, the search range at these beginning vertices is enlarged. When $k = 50$, scanning of all the data points is unavoidable. Note that since the candidate points found is always less than 100, the skyline operation does not have a substantial impact on the performance.

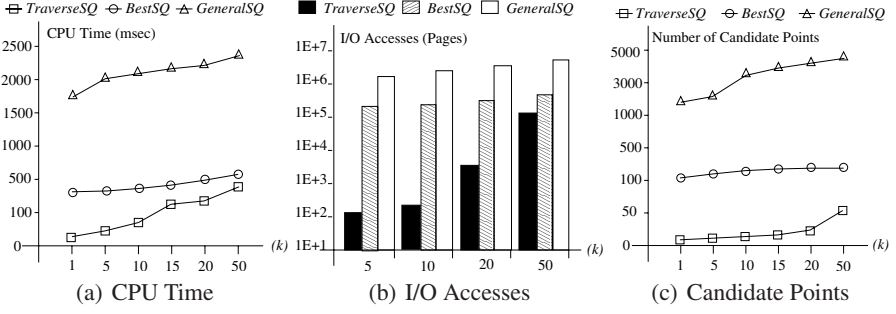


Fig. 6. Performance of Algorithms Versus k .

As k is increased, the cost of the *BestSQ* algorithm increases slightly. This is because when k increases, the cost of searching for the k th nearest neighbor increases. But the cost of the two range queries does not increase much since the route size is the major factor in determining the range size. Also note that since the maximum number of candidate points for *BestSQ* is 279, when k is bigger than 15, the amount of candidate points comes to be this maximum value.

The cost of the *GeneralSQ* algorithm also increases with an increase in k . Since the route size is fixed at 100 in this experiment, the number of range queries is constant. The bigger k is, the larger the size of these range queries. The slight increase in cost as k increases indicates that route size has the biggest effect on performance of *GeneralSQ*. Also, note that as the number of candidate points is always below 5,000, the skyline operation does not have great impact on the overall cost.

5.3 Experiment on the Effect of Buffer Size

In this experiment, k is set to 5 and the route size is set to 100. The buffer size is varied from 10% to 50%. Since the buffer size has little influence on the CPU time and number of candidate points, the experiment only considers I/O cost. It can be seen in Figure 7 that the number pages accessed decreases greatly with as the buffer size increases. This is because when the buffer grows, more road network data reside in the buffer, thus decreasing the chances of reading data from outside the buffer. For the *TraverseSQ* algorithm, when the buffer size increases to 20% of the road network, the I/O access do not change since the size of the pages accessed by the *TraverseSQ* algorithm is smaller than the size of the buffer.

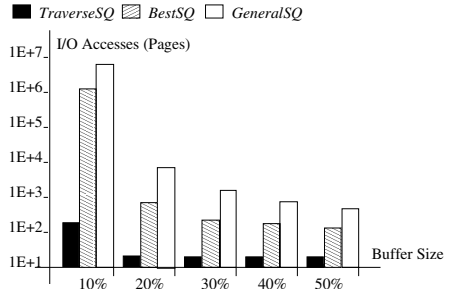


Fig. 7. I/O Accesses Versus Buffer Size.

6 Summary and Future Work

This paper introduces a novel location-based query in a road network based usage scenario, namely the in-route nearest neighbor skyline query. Two interesting special case of the general usage scenario considered are identified. The paper then proceeds to reuse and extend existing query processing techniques to apply to the new type of query. Specifically, it provides algorithms for the two special cases as well as the general case. Finally, the paper reports on experimental performance studies with the three algorithms. Real point of interest and road network data are used.

Providing efficient support for in-route location-based queries for moving users is an interesting and important topic in traveler information system. Since movement is normally restricted to a transportation network, traditional spatial-temporal queries, e.g., nearest neighbor, spatial range, closest pair, distance join, need to be re-considered because Euclidean distance becomes inappropriate. The algorithms proposed in this paper may be extended to make use of indexing and pre-computation techniques. Several research directions may be identified, including the following two:

- It is required in this paper that the pre-defined route is a sequence of neighboring vertices, while in the real world a pre-defined route may also consist simply of several specified locations in the road network. Algorithms in the context of such routes is an interesting direction for future work.
- In this paper, all the data points are organized into one group, and the in-route query finds k candidate points from this group. More complex query preferences may occur naturally in real applications. For example, a moving user may want to visit a bank as well as a supermarket before arriving at the destination. Processing of location-based queries under such complex settings is also an interesting direction for future work.

Acknowledgments

This work was supported in part by grant 216 from the Danish National Center for IT Research. In addition to his primary affiliation, the second author is an adjunct professor at Agder University College, Norway.

References

1. R. Benetis, C. S. Jensen, G. Karčiauskas, S. Šaltenis. Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects. In *Proc. IDEAS*, pp. 44–53, 2002.
2. S. Borzsonyi, D. Kossmann, K. Stocker. The Skyline Operator. In *Proc. ICDE*, pp. 421–430, 2001.
3. A. Brilingaitė, C. S. Jensen, N. Zokaitė. Enabling Routes as Context in Mobile Services. In *Proc. ACM GIS*, pp. 127–136, 2004.
4. J. Chomicki, P. Godfrey, J. Gryz, D. Liang. Skyline with Presorting. In *Proc. ICDE*, pp. 717–816, 2003.
5. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to Algorithms: Second Edition*. The MIT Press, 2001.

6. E. Frentzos. Indexing Objects Moving on Fixed Networks. In *Proc. SSTD*, pp. 289–305, 2003.
7. A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proc. SIGMOD*, pp. 47–57, 1984.
8. C. Hage, C. S. Jensen, T. B. Pedersen, L. Speičys, and I. Timko. Integrated Data Management for Mobile Services in the Real World. In *Proc. VLDB*, pp. 1019–1030, 2003.
9. C. S. Jensen, J. Kolář, T. B. Pedersen, I. Timko. Nearest Neighbor Queries in Road Networks. In *Proc. ACMGIS*, pp. 1–8, 2003.
10. D. E. Knuth. *Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley Pub Co., 1998.
11. D. Kossmann, F. Ramsak, S. Rost. Shooting Stars in the Sky: an Online Algorithm for Skyline Queries. In *Proc. VLDB*, pp. 275–286, 2002.
12. H. X. Lu, Y. Luo, X. Lin. An Optimal Divide-Conquer Algorithm for 2D Skyline Queries. In *Proc. ADBIS*, pp. 46–60, 2003.
13. D. Pfoser, C. S. Jensen. Indexing of Network Constrained Moving Objects. In *Proc. ACMGIS*, pp. 25–32, 2003.
14. D. Papadias, Y. Tao, G. Fu, B. Seeger. An Optimal and Progressive Algorithm for Skyline Queries. In *Proc. SIGMOD Conf.*, pp. 467–478, 2003.
15. D. Papadias, J. Zhang, N. Mamoulis, Y. Tao. Query Processing in Spatial Network Databases. In *Proc. VLDB*, pp. 802–813, 2003.
16. N. Roussopoulos, S. Kelley, F. Vincent. Nearest Neighbor Queries. In *Proc. SIGMOD*, pp. 71–79, 1995.
17. L. Speičys, C. S. Jensen, A. Kligys. Computational Data Modeling for Network Constrained Moving Objects. In *Proc. ACMGIS*, pp. 118–125, 2003.
18. C. Shahabi, M. R. Kolahdouzan, M. Sharifzadeh. A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases. In *GeoInformatica* 7(3), pp. 255–273, 2003.
19. Z. Song, N. Roussopoulos. K-Nearest Neighbor Search for Moving Query Point. In *Proc. SSTD*, pp. 79–96, 2001.
20. S. Shekhar, J. S. Yoo. Processing In-Route Nearest Neighbor Queries: A Comparison of Alternative Approaches. In *Proc. ACMGIS*, pp. 9–16, 2003.
21. K. L. Tan, P. K. Eng, B. C. Ooi. Efficient Progressive Skyline Computation. In *Proc. VLDB*, pp. 301–310, 2001.
22. Y. Tao, D. Papadias, Q. Shen. Continuous Nearest Neighbor Search. In *Proc. VLDB*, pp. 287–298, 2002.

P2P Spatial Query Processing by Delaunay Triangulation

Hye-Young Kang, Bog-Ja Lim, and Ki-Joune Li

Department of Computer Science and Engineering
Pusan National University, Pusan 609-735, South Korea
{hykang,bjlim,lik}@isel.cs.pusan.ac.kr

Abstract. Although a number of methods have been proposed to process exact match or range queries in P2P network, very few attention has been paid on spatial query process. In this paper, we propose a triangular topology of P2P network and spatial query processing methods on this topology. Each node maintains a set of pointers to neighbor nodes determined by delaunay triangulation. This triangular network topology provides us with an efficient way to find a path from any source node to the destination node or area, and to process spatial queries. We compare two spatial query processing methods based on the triangular network by experiments.

1 Introduction

With the recent technical progress of wireless communications, sensor networks, and location tracking systems such as GPS, P2P becomes an important approach of massively distributed systems not only for file transfers but also for searchable querical data network [1]. In order to enlarge P2P application area, we need to broaden the types of query processing and improve the performance. For this reason, a number of searching and indexing methods have been proposed including distributed hashing table such as CAN [2], and Chord [3]. But they are limited to exact match search or keyword search. When we consider a set of P2P nodes as a massively distributed database, several types of search functions should be provided in addition to exact match search. Some methods for processing range queries have been also proposed to meet this functional requirements [4][5].

P2P technique can be applied to handle data distributed over a large number of nodes in sensor network. In particular, it is an important approach to process spatial query, where nodes in sensor networks are equipped with GPS. Suppose that a query such as "Find the average temperature within 100 meters from point A", is submitted to a node in this type of sensor network. If the number of nodes exceeds a certain limit, in-network processing or P2P technique are more scalable than centralized approach. By P2P technique, a subset of nodes are to be engaged in to process this query, starting from the node without submitting the query to central server. However, very few attention has been paid on spatial query processing by P2P approach according to our knowledge. The goal of our work is to develop an efficient method of spatial query processing.

In general, spatial query processing has two phases; routing phase and refinement phase as depicted by figure 1. The routing phase is to forward the submitted query to an area where the spatial query condition is satisfied. The nodes in this area belong to candidates of the spatial query. And the refinement phase is to check every node in the area to evaluate the query.

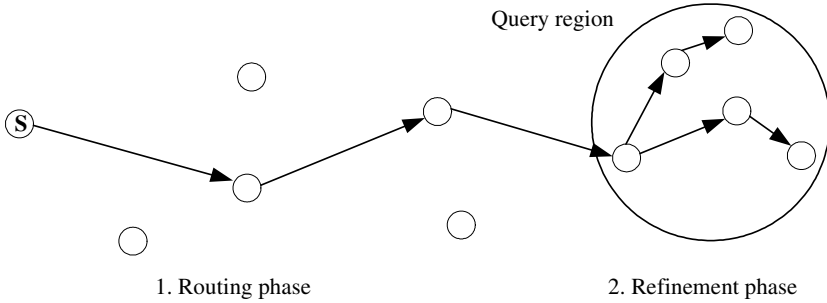


Fig. 1. Example of routing and refinement phases.

A certain number of geographic routing methods have been developed for ad-hoc networks environment. These methods provide with various routing methods from a source node to destination node. These routing methods are analogous with the routing phase mentioned in the previous paragraph. But they are based on a geographic property of ad-hoc network that the nodes receiving the messages from a source node are within a certain distance from the source node. It means that the neighbor nodes are in the broadcasting coverage from the source node. Most routing methods for ad-hoc networks find a routing path by forwarding message to the neighbor nodes in the broadcasting coverage. In infrastructure network, we are however unable to find the neighbor nodes of a given node without explicit pointers, such as IP address. As a consequence, the routing methods for ad-hoc networks cannot be used for infrastructure networks.

In this paper, we propose a method, which explicitly specifies the neighbor nodes of each node instead of broadcasting area of ad-hoc network like DHT such as finger table in Chord [3] or neighbor grid pointer in CAN [2]. And we find a routing path from the source node to the destination with neighbor node pointer as DHT. But our method differs from DHT in that node pointers of DHT have no geographic sense. Our method determines the neighbor nodes by delaunay triangulation. Then the routing path of our method consists of edges in triangular networks. The triangulation is performed in an incremental and distributed way without any centralized data structure since we cannot maintain any kind of centralized data in P2P environment.

This paper is organized as follows. In section 2, we introduce related work and the motivation of our work. And we propose a triangular network topology and spatial query processing algorithms for two types of query in P2P environment,

which are nearest neighbor query and range query in section 3. We present the results of experiments with our method in section 4, and conclude our paper in section 5.

2 Related Work

A simple way to find a given data in P2P environment is to flooding messages over the networks [6]. But this method results in a large number of messages and a traffic overhead. DHTs has been proposed by several methods such as Chord [3] and CAN [2] to reduce traffic overhead and hop counts from the source node to the destination. By Chord, the query message can be reached to the destination within $O(\log N)$ hop counts by using finger table, while CAN needs $O(N^{1/d})$ hop counts where N and d means the number of nodes and the dimensionality of transformed space of CAN. Other methods such as Pastry [7], Tapestry [8], and Bristle [9] are based on DHT although their DHTs and topologies are different.

While DHT is useful to process exact match or keyword match queries, it cannot be used in range query process. In order to overcome this problem of DHT, several extensions have been made. For example, an extension of CAN [4] is proposed to handle range query, and Chord is extended to process range query by [5]. In contrast with these methods, PePeR(Peer to Peer Range) [10] does not use any kind of DHT but distributes records into nodes according to their attribute values and maintains links to the nodes with adjacent values. The links provide routing path to search a given query range in P2P environment. An interesting method has been also proposed to process multi-dimensional query by [5].

However none of these methods except [11] and [12] are capable of processing spatial query in P2P environment, where spatial data are spread over nodes.

While each node explicitly specifies its links and the routing from a source node to destination is carried out via the link information in P2P environment, neighbor nodes are not explicitly described in ad-hoc networks. If a message from a node is received by another node in ad-hoc networks, it implies that they are closely located. Spatial queries are processed without any explicit links like P2P environment. A number of routing algorithms to find routing paths from a source node to a geographically specified destination. For example, LAR [13] has been proposed to find geographic routing path without excessively flooding messages to neighbor nodes in ad-hoc networks. And several routing algorithms for ad-hoc networks have been proposed such as GPSR(Greedy Perimeter Stateless Routing) [14], *Spatial Query Routing* and *Pipeline Forwarding* [15]. In particular, these methods assume the mobility of nodes.

But these methods cannot be used in P2P environment, since they do not maintain link information in explicit way, whereas routing in P2P is carried out via traversing links between nodes. In order to apply one of these methods to P2P environment, we should determine how to select and maintain neighbor nodes, which are not explicitly specified in ad-hoc networks.

The goals of this paper are therefore, firstly to provide with a scheme to select geographically neighbor nodes and maintain links to them for P2P environment and secondly to develop routing and spatial query processing methods with the link information.

3 Triangular Network Topology

As mentioned in the previous section, geographical link information should be maintained by each node to process spatial query. In this section, we propose a triangular network topology, which explicitly specifies neighbor nodes with edges of triangular network. In order to establish triangular networks, we use delaunay triangulation [16], since it has some important geometric properties, which are crucial in processing spatial queries.

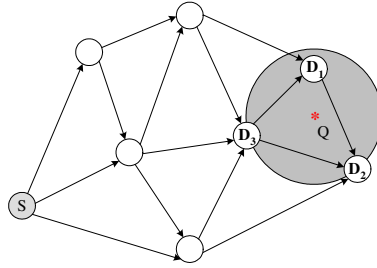


Fig. 2. Example of spatial query processing by triangular network.

Figure 2 shows an example of using delaunay triangulation for spatial query processing in P2P environment. Suppose that a query is submitted to node S to find the nearest node from a point Q . In this example, each node maintains links to neighbor nodes specified by edges of triangles. And starting from node S , we can forward query message to the triangle $\triangle D_1D_2D_3$ that contains point Q . Then the nearest node must be one of D_1 , D_2 , and D_3 according to the properties of delaunay triangulation.

In order to realize this method of query processing, we should consider the following issues.

- Distributed delaunay triangulation : in P2P environment, no centralized data structure is maintained, and every processing must be performed in a distributed way.
- Spatial query processing in triangular networks : the query processing with triangular network consists of two phases; *routing phase* and *refinement phase*. The routing phase is to forward query message to the specified query region where candidate nodes are located. This phase is performed via traversing edges of triangles. And the refinement phase is to select the nodes satisfying query condition among the candidate nodes.

We will propose solutions to each issues in the subsequent subsections. But these solutions are based on the following assumptions.

- All nodes are stationary.
- An IP address is allocated to each node. It may be unpractical in IPv4 environment, but realistic in IPv6.
- Each node has the following information;
 - its location,
 - its IP address, and
 - the locations and IP address of neighbor nodes linked via triangular networks.

3.1 Routing Algorithms

In this subsection, we will propose two routing methods with triangular networks. The goal of routing algorithm is to forward spatial query message starting from the source node to a region where the spatial query predicate is satisfied. The two routing methods we propose in this papers differ in how to decide the nodes to forward query message.

The first method, called *half-moon method*, forwards query message to the nodes located within the half-moon defined as depicted by figure 3. The flooding is limited by the half-moon.

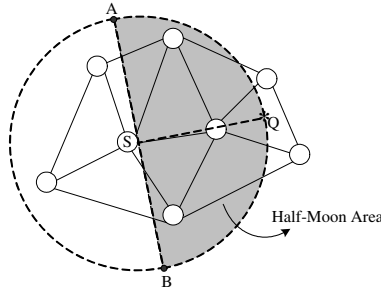


Fig. 3. Example of half-moon.

Definition 1. Let point S and D be the position of source node and the center of query area respectively. Then the half-moon of source S and destination D is the half circle with center S , radius $r = |SD|$, containing three points A , B , and d where A and B are on the perimeter of the circle so that $\overline{AB} \perp \overline{SD}$.

Figure 4 shows the half-moon method, where a query is submitted to a node s . Since node N_1 , N_2 , and N_3 are contained by the half-moon of source S and destination Q , the message is forwarded to them. The arrows of figure 4 mean the message forwarding by half-moon method. This forwarding is repeated until the message is reached to the query region, which is the shaded circle in figure 4.

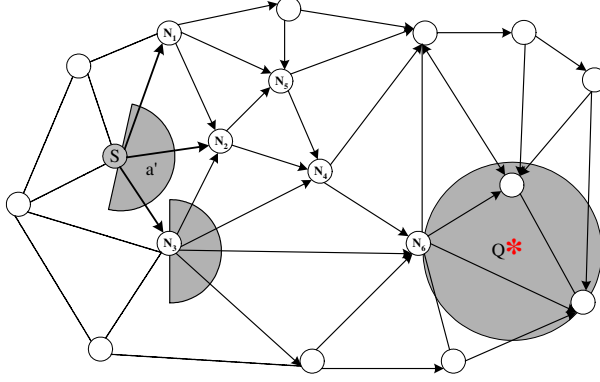


Fig. 4. Example of half-moon routing method.

By the half-moon routing method, we find a routing path with relatively small hop counts, while the number of flooded messages are considerable. We propose the second routing method, which gives a routing path with more hop counts but produces a small number of messages. We call this method *greedy triangular routing method*.

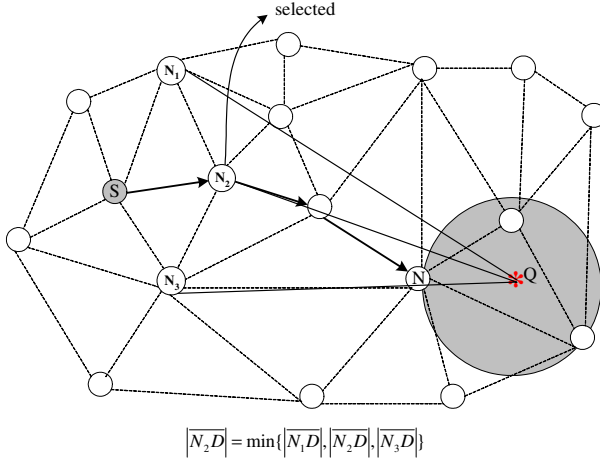


Fig. 5. Example of greedy triangular routing method.

While query message is forwarded to every node in the half-moon of source node by the half-moon method, only one node is selected to forward query message by greedy triangular method. This node N_k is selected among the nodes N_i linked to the source node, so that $|N_kQ| = \min(|N_iQ|)$. Figure 5 shows an example of routing by greedy triangular method.

In comparison with the half-moon method, the greedy triangular routing method gives relatively long routing path between the source node and the query region. But it results in a very small number of messages for routing. Detail comparisons will be made in section 4 with experiment results.

3.2 Refinement

After reaching at the query region, the nodes in the query region spatial query condition must be carefully examined to see if they belong to the spatial query result. In this paper, we propose refinement methods for two types of queries; nearest neighbor and range query. For reason of simplicity, we propose a refinement algorithm where the query region is given as a circle. But this algorithm can be easily extended to any type query shape.

The query message is forwarded to one of nodes contained by query region. Then the properties of delaunay triangulation allow to find the rest nodes contained by the region with ease. Figure 6 explains how to carry out the refinement.

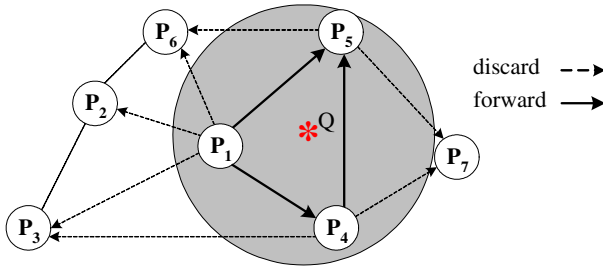


Fig. 6. An example of refinement for range query.

As shown in figure 6, the query message is forwarded to the neighbor nodes, and if neighbors are no longer within the query region. And since all nodes within the query region are connected to the firstly visited node, they are accessible from the first node. Algorithm 1 summaries this procedure.

This algorithm is correct since we have proven the correctness based on the properties of delaunay triangulation. The theorem and proof are given as follows.

Theorem 1. Refinement of circle range query

If a node P_1 is within a circle $R(q, r)$, where q and r are the center and radius respectively. Then all nodes within circle R can be found by algorithm 1 starting from node P_1 .

Proof. Here, we prove that if a circle $R(q, r)$ only contains a node p_1 but all neighbor nodes of p_1 is out of $R(q, r)$, there is no node in $R(q, r)$ but p_1 . So, we assume that a node p_4 is not a neighbor node of p_1 and is within $R(q, r)$. We use some notation for proof as following.

Algorithm Range query

Input. Node n , Query range w
 Ouput. Node $R[]$
Begin
 queue $Q \leftarrow \phi$
 $R \leftarrow \phi$
 $R \leftarrow R \cup \{n\}$
 insert neighbor nodes of n to Q
while Q is not empty {
 $x \leftarrow \text{Dequeue}(Q)$
 if x is within w **then** {
 $R \leftarrow R \cup \{x\}$
 insert neighbor nodes of x to Q
 }
}
End

Fig. 7. Algorithm 1 : Refinement of range query.

- $\triangle p_1 p_2 p_3$ is the the nearest triangle from p_4 and includes p_1 .
- $C(\triangle p_1 p_2 p_3)$ is a circumcircle of $\triangle p_1 p_2 p_3$.
- n is a intersection point between $C(\triangle p_1 p_2 p_3)$ and $\overline{p_1 p_4}$.
- S is a minimum circle including p_1 and p_4 .
- $\widehat{p_1 p_2 p_3}$ is a arc consisting of p_1, p_2, p_3 . A node $p_i (i=1,2,3)$ is on a perimeter of circle.
- $\widehat{p_1 p_i n} (i=2,3)$ is $\min(|\widehat{p_1 p_2 n}|, |\widehat{p_1 p_3 n}|)$.

A circle R always contain a circle S . And a circle S contains a arc $\widehat{p_1 p_i n} (i=2,3)$. A node p_2 or p_3 is a neighbor node of p_1 . Therefore, when a node p_4 is in R , there is at least on neighbor node of p_1 in R .

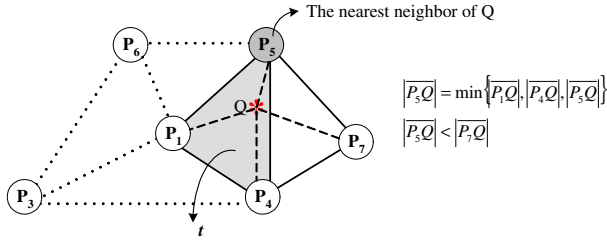
The algorithm for the refinement of nearest neighbor query slightly differs from that for region query. By routing from the source node, we can get the triangle T containing the query point q . According to the properties of delaunay triangulation, the nearest neighbor node is one of the vertices of triangle T or the adjacent triangles of T . Algorithm 2 explains this procedure.

The correctness of this algorithm is shown by the following theorem and figure 9 depicts the procedure.

Theorem 2. Refinement of nearest neighbor query

The nearest neighbor node to point q is one of three vertices of triangle containing q or the vertex of the triangle sharing the nearest to q .

Proof. A point q is in a triangle t consisting of $\{n_1, n_2, n_3\}$. A $C(\triangle n_1 n_2 n_3)$ is a circumcircle of the triangle t . d_i is a distance between query point q and n_i . The nearest edge from q at t is $near_edge$ and a triangle t' is the triangle sharing $near_edge$ with t .

Algorithm Nearest Neighbor QueryInput. Query point q , triangle $t\{p_1, p_2, p_3\}$ containing q Output. Nearest neighbor node n **Begin**Find p_i such that $\forall i, j \quad |\overline{qp_i}| < |\overline{qp_j}| \quad (i \neq j)_{(1 \leq i, j \leq 3)}$ $n \leftarrow p_i$ Find node x of triangle sharing the nearest edge of t from q **if** $|\overline{xq}| < |\overline{nq}|$ **then** $n \leftarrow x$ **return** n **End****Fig. 8.** Algorithm 2: Refinement of nearest neighbor query.**Fig. 9.** An example of refinement for nearest neighbor query.

If a distance d_1 is equal to d_2 and d_3 then, d_1 is equal to radius of $C(\triangle n_1, n_2, n_3)$. So, According to the delaunay triangle property, distances between q and other nodes is longer than $d_i (1 \leq i \leq 3)$.

If each distance is not equal, we can also prune candidates of nearest neighbor node. We can get the shortest distance, $dist$, among $d_i (1 \leq i \leq 3)$. We can consider a intersection point, poi , of a circle $S(q, dist)$ and $C(\triangle n_1, n_2, n_3)$. A point poi always is on a arc of $near_edge$. Therefore, distances between one node of triangles sharing edges of t and q is always longer than $dist$ except the triangle t' .

3.3 Join: Incremental Growth of Triangular Network

In order to maintain triangular networks, we need a join algorithm for incremental growth, since we have no centralized data structure in P2P environment. It means that the delaunay triangulation should be performed in fully distributed way, when a new node joins the network.

The join algorithm is simple and easy to implement by using the nearest query processing algorithm mentioned in the previous subsection. The joining procedure is composed of two steps;

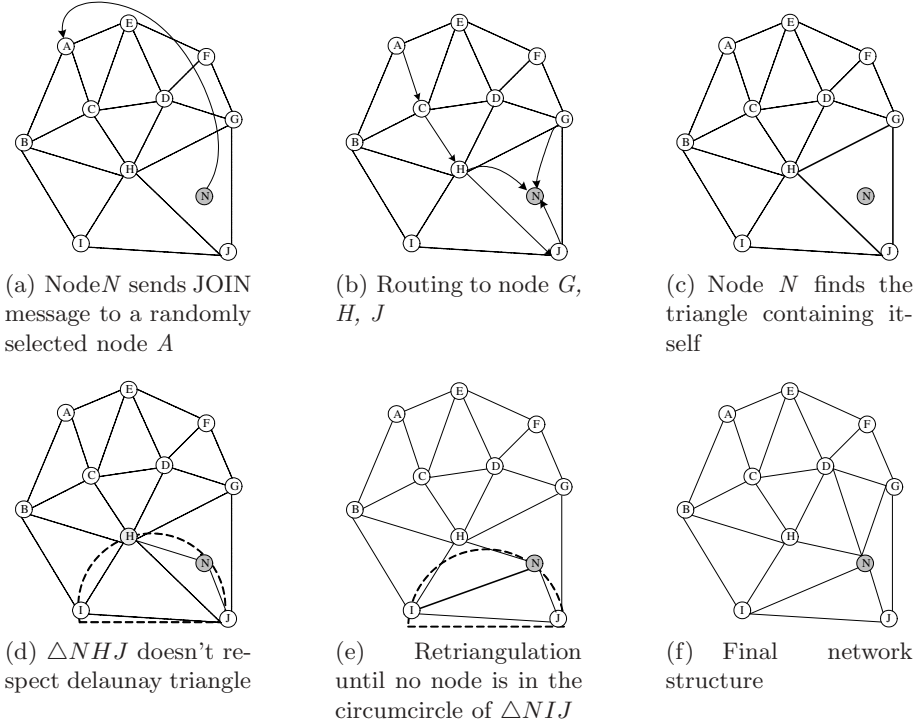


Fig. 10. Example of incremental growth of triangular network.

- **step 1: finding the triangle containing the new node**
- **step 2: local modification of Delaunay triangles**

Figure 10 shows an example of join procedure. For step 1, the new node sends a JOIN message to a randomly selected or predefined node. We suppose that any new node knows the IP addresses of at least one node before joining the network. Then by nearest neighbor query processing method, we can find the nearest node to the new node. And starting from the nearest neighbor node, we can find the triangle containing the new node.

And for step 2, we insert three edges from the new node to each node of the triangle containing the new node in order to make new triangles. If the new triangles do not respect the condition of Delaunay triangulation, we should modify the triangles as illustrated by figure 10(e) and 10(f). The algorithm of join operation is summarized by figure 11.

4 Experiments

We performed experiments to observe the performance of our methods and compare two routing strategies by simulation. Three data sets with 10,000 nodes are synthetically generated with different distributions; uniform, clustered, and

Algorithm Node Join

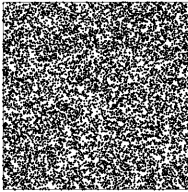
```

Input.   new node  $n$ 
Output.
Begin
  /*  $DT$  : Delaunay Triangulation;
    $NL$  : Neighbor node List;
    $CCW(\overline{ab})$ : next count-clockwise neighbor node of  $\overline{ab}$  at node  $a$ 
    $C(\triangle abc)$  : circumcircle of  $\triangle abc$ 
  */
  Find  $p_1, p_2, p_3$  such that  $\triangle p_1 p_2 p_3$  is  $DT$  ,
   $\forall i, j \ p_i \in p_j.NL (i \neq j)$  and  $\triangle p_1 p_2 p_3$  contains  $n$ 
  for each node  $p_i (1 \leq i \leq 3)$  {
     $x \leftarrow CCW(\overline{np_i})$ 
     $x.NL \leftarrow x.NL \cup n$ 
     $n.NL \leftarrow n.NL \cup x$ 
    while(  $\exists y, C(\triangle np_i x)$  contain  $y$ ) {
       $p_i.NL \leftarrow p_i.NL - x$ 
       $x.NL \leftarrow x.NL - p_i$ 
       $x \leftarrow CCW(\overline{np_i})$ 
       $n.NL \leftarrow n.NL \cup x$ 
    }
  }
End

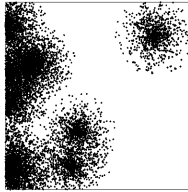
```

Fig. 11. Algorithm of Node join.

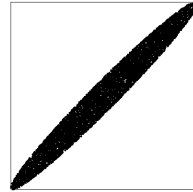
skewed distributions as shown by figure 12. We observed the query processing performance for range query and nearest neighbor query. First, figure 13 shows the performances of region query processing. We computed the average numbers of messages, engaged nodes, and routing hops, respectively, to process 1000 randomly generated queries. As we expect, the number of messages by the half-moon method is significantly larger than that of greedy triangular method. Note that the y -axis in figure 13 is log-scaled. And the number of message is almost the number of total nodes. It means that the messages are flooded over most of nodes and consequently cause a serious communication overhead. And the graph



(a) Uniform



(b) Clustered

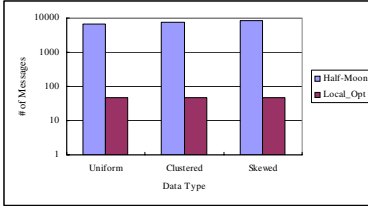


(c) Skewed

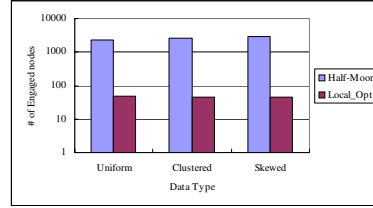
Fig. 12. Distributions of data sets.

Table 1. Experiment results for region query.

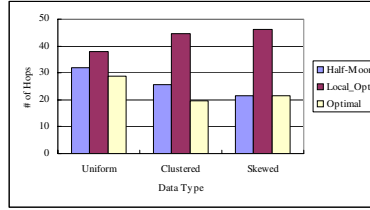
Measure	Method	Uniform	Clustered	Skewed
# of Messages	Half-moon	6734.55	7667.46	8476.32
	Local_Opt	46.7	46.64	47.36
# of Engaged Nodes	Half-moon	2289.3	2594.86	2857.71
	Local_Opt	47.23	45.63	46.46
# of Hops	Half-moon	32.04	25.71	21.37
	Local_Opt	37.89	44.73	46.06
	Optimal	28.68	19.65	21.52



(a) Number of Messages



(b) Number of Engaged Nodes



(c) Hop count

Fig. 13. Experiment results for region query.

for the number of engaged nodes in figure 13(b) is analogous with the graph in figure 13(a).

On the contrary, the half-moon method gives very good routing paths in comparison with the greed triangular method. And it is slightly worse than the optimal routing path. But the hop count of routing path affects the response time and it is in fact not a critical factor, since the time to forward message is negligible.

The results of nearest neighbor query are almost same with those for region query. The reason is that the performances are mainly determined by the routing phase rather than the refinement phase, and the routing methods for nearest neighbor and region query are identical except their termination conditions.

5 Conclusion

A number of searching mechanisms have been proposed for exact match and range queries, very little attention have been on spatial query processing method

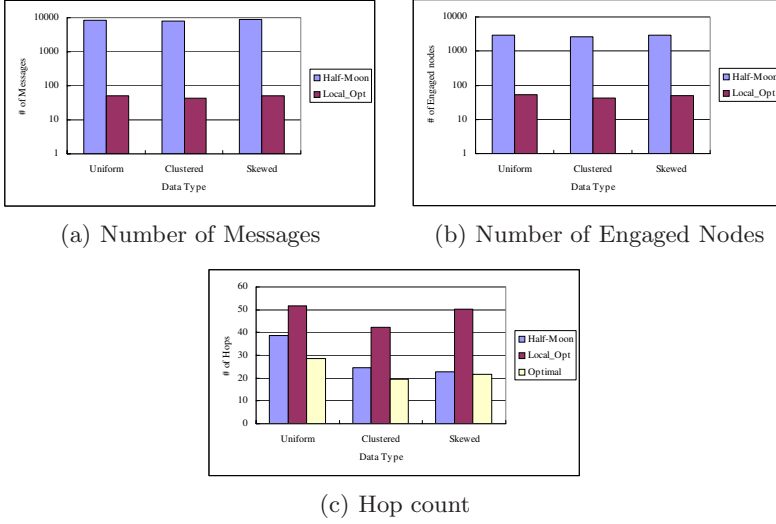


Fig. 14. Experiment results for nearest neighbor query.

Table 2. Experiment results for nearest neighbor query.

Measure	Method	Uniform	Clustered	Skewed
# of Messages	Half-moon	8626.53	7821.98	8702.01
	Local_Opt	51.78	42.14	50.1
# of Engaged Nodes	Half-moon	2925.47	2643.99	2933.56
	Local_Opt	52.78	43.14	51.1
# of Hops	Half-moon	38.51	24.71	22.73
	Local_Opt	51.78	42.14	50.1
	Optimal	28.68	19.65	21.52

in P2P environment. This paper has two major contributions; 1) we proposed a triangular network topology to process spatial queries in P2P environment, where a node has links to neighbors via edges of triangular network, and 2) spatial query processing methods have been proposed with the triangular network topology.

First, we apply delaunay triangulation to maintain a triangular network, since it has some useful properties in processing spatial queries. And it is performed in an incremental way due to the lack of centralized data structure in P2P network.

Second, the spatial queries are performed in two phases by our method; routing phase and refinement phase. In routing phase, the query message is forwarded to the query region from source node via edges of triangles. We proposed two strategies for finding routing paths. And in refinement phase, the spatial query condition must be evaluated. We proposed several query processing algorithms for the refinement phase based on the properties of delaunay triangulations.

Future work includes improvements of routing methods. While the half-moon routing method gives a good routing path, it results in an excessive flooding of messages. On the contrary, greedy triangular routing method gives relatively long routing path but produces a small number of messages. But the hop count of the routing path and the number of forwarded messages are both important performance measures. We should therefore find a compromise between two approaches without sacrificing one measure for the other.

Acknowledgement

This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce, Industry and Energy of the Korean Government.

References

1. Farnoush Banaei Kashani and Cyrus Shahabi. Searchable querical data networks. In *Databases, Information Systems, and Peer-to-Peer Computing*, pages 17–32, 2003.
2. Mark Handley Richard Karp Sylvia Ratnasamy, Paul Francis and Scott Schenker. A scalable content-addressable network. In *ACM SIGCOMM Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, 2001.
3. David Karger M. Frans Kaashoek Ion Stoica, Robert Morris and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, 2001.
4. Artur Andrzejak and Zhichen Xu. Scalable, efficient range queries for grid information services. In *Proceedings of the Conference on Peer-to-Peer Computing*, page 33, 2002.
5. Jinbo Chen Min Cai, Martin Frank and Pedro Szekely. Maan: A multi-attribute addressable network for grid information services. In *Proceedings of Grid Computing*, page 184, 2003.
6. Gnutella. In <http://gnutella.wego.com/>.
7. Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware, IFIP/ACM Conference on Distributed Systems Platforms*, pages 329–350, 2001.
8. John D. Kubiawicz Ben Y. Zhao and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. In *Technical Report, University of California at Berkeley*, 2001.
9. Chung-Ta King Hung-Chang Hsiao. Bristle: A mobile structured peer-to-peer architecture. In *Proceedings of Symposium on Parallel and Distributed Processing*, page 33.1, 2003.
10. Xinghua An Antonios Daskos, Shahram Ghandeharizadeh. Peper: A distributed range addressing space for peer-to-peer systems. In *Proceedings of Databases, Information Systems, and Peer-to-Peer Computing*, pages 200–218, 2003.
11. Filipe Araujo and Luis Rodrigues. Geopeer: A location-aware peer-to-peer system. DI/FCUL TR 03–31, Department of Informatics, University of Lisbon, December 2003.

12. Michael Nahas Jorg Liebeherr and Weisheng Si. Application-layer multicasting with delaunay triangulation overlays. UVa/CS TR 01-26, Department of Computer Science, University of Virginia, December 2001.
13. Young Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *Proceedings of Mobile computing and networking*, pages 66-75, 1998.
14. Brad Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of Mobile computing and networking*, pages 243-254, 2000.
15. Yingqi Xu and Wang-Chien Lee. Window query processing in highly dynamic sensor networks: Issues and solutions. In *Proceedings of GeoSensor Networks*, 2003.
16. Franco.P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction*. Springer Verlag, 1985.

Expansion-Based Algorithms for Finding Single Pair Shortest Path on Surface

Ke Deng and Xiaofang Zhou

School of Information Technology and Electrical Engineering
University of Queensland, Brisbane, QLD 4072 Australia
{dengke,zxf}@itee.uq.edu.au

Abstract. Finding single pair shortest paths on surface is a fundamental problem in various domains, like Geographic Information Systems (GIS) 3D applications, robotic path planning system, and surface nearest neighbor query in spatial database, etc. Currently, to solve the problem, existing algorithms must traverse the entire polyhedral surface. With the rapid advance in areas like Global Positioning System (GPS), Computer Aided Design (CAD) systems and laser range scanner, surface models are becoming more and more complex. It is not uncommon that a surface model contains millions of polygons. The single pair shortest path problem is getting harder and harder to solve. Based on the observation that the single pair shortest path is in the locality, we propose in this paper efficient methods by excluding part of the surface model without considering them in the search process. Three novel expansion-based algorithms are proposed, namely, Naïve algorithm, Rectangle-based Algorithm and Ellipse-based Algorithm. Each algorithm uses a two-step approach to find the shortest path. (1) compute an initial local path. (2) use the value of this initial path to select a search region, in which the global shortest path exists. The search process terminates once the global optimum criteria are satisfied. By reducing the searching region, the performance is improved dramatically in most cases.

1 Introduction

The algorithms for finding shortest path on polyhedral surface are the fundamental in a large variety of applications and research areas. In robotic systems, finding an optimal collision-free path for robots in a given space [8] is known as path planning. With the path planning technology, the virtual camera in virtual endoscopy can be guided to pass through an anatomical object for comprehensive disease diagnosis [5]. In 3D GIS systems, many applications require reporting the shortest path between two entities over terrain as a basic function. In bioinformatics, by searching and comparing shortest paths between objects, tasks like gene recognition and protein structure analysis could be conducted. In addition, another emerging interesting research area is the surface nearest neighbor query in spatial database, which is essential in applications such as emergency response, wildlife behavior monitoring and battlefield surveillance.

Many algorithms have been proposed to find the shortest path on surfaces. These algorithms usually work on polyhedral surfaces which are represented as meshes consisting of triangular faces. Meshes are usually generated by sampling and triangulating uniformly distributed points on the surface of the target object. For example, in GIS, the terrain surface is normally represented as the triangular network for visualization or other purpose. Such network can be either regular or irregular. In principle, existing shortest path algorithms try to solve three major problems: all pairs, single source and single pair. While the all pairs problem is to find the shortest paths between any pair of vertices of the triangulated mesh, the single source problem is to find the shortest paths from the fixed source point to any other vertices. The single pair problem is to find the shortest path from the fixed source point to some specified destination point. In this paper, we will address the single pair problem on the mesh surface. Efficient algorithm for single pair problem has special interest in applications like surface KNN and path planning.

Among those algorithms, some of them [7–9] first pre-process the polyhedral surface into subdivisions, whereby the exact shortest path from the fixed source point to a given query point can be reported quickly. Chen & Han proposed an algorithm [2, 3], which is one of the best and the only feasible algorithm today. Without any pre-processing, Chen & Han algorithm can find the exact shortest path in time complexity $O(n^2)$. To search the global optimal shortest path, all these algorithms need to traverse the entire polyhedral surface. On the other hand, advances in areas such as GPS, CAD systems and laser range scanner are producing growing larger polyhedral surface datasets. A surface containing millions of polygons are not uncommon. Typically, processing the geometric data is very computational expensive and memory hungry. This makes the single pair problem extremely costly and less practical even with high profile computer system. It is urgent to develop approaches to alleviate the situation.

Some works have been done. Takashi et. al. [4] proposed an algorithm for the approximate shortest path with selective refinement technology on polyhedral surface. Dinesh et. al. [6] developed an algorithm based on multiresolution technology. Both of them improved the performance of single-pair problem notably. However, their algorithms still need to traverse the entire surface. The initiative behind these methods is to reduce the computational complexity by reducing the problem complexity. Based on the similar concept, we develop some new algorithms.

We observed that the local shortest path connecting two points inside a certain region is also global optimal. For example, the shortest path between Brisbane and Sydney cannot be the one that pass through Melbourne. Motivated by this observation, our proposed algorithms use a two-step approach to find the shortest path. First, we need to find an initial local path. Second, we use the value of this initial path to select a search region, in which the global shortest path exists. In this paper, three algorithms: Naïve algorithm, Rectangle Border algorithm (RB) and Ellipse Border algorithm (EB) are developed. To our knowledge, no such algorithms have been proposed before. The Naïve

algorithm is conceptually simple and works as the benchmark to evaluate the other two algorithms. In RB algorithm, we offer two alternatives terms of the way of testing global optimum criteria and extension methods, namely Rectangle Border Direct Algorithm (RBD) and Rectangle Border Indirect Algorithm (RBID). In EB algorithm, heuristics is used to estimate the initial length with the consideration of surface roughness.

Comparing with others, our algorithms have four advantages. First, they do not require traversing the entire polyhedral surface. By narrowing down the search region to a selected area, the processing time can be significantly reduced, hence improve the performance of existing algorithms. Second, it is easy to graft our technique on top of an existing algorithms. They can work with existing shortest path algorithms, such as Chen & Han algorithm and Takashi et. al. approximate algorithm. Third, our algorithms are simple and easy to implement. Last, the idea used in our algorithm is novel. Two steps in our proposed algorithms are independent each other. Several different methods are presented for each step in this paper. This provides a great flexibility to select an optimum combination for the best performance in some context.

The rest of the paper is organized as follows. In Sect. 2, a brief overview of related works is presented. Section 3 describes our three proposed algorithms in detail. Performance study is discussed in Sect. 4. Finally, conclusions are given in Sect. 5.

2 Related Work

Shortest path problems have been studied for the last two decades. For a given polyhedral surface and a point, the surface can be pre-processed to produce a data structure. The shortest path from the specified point to any query point can be reported with the aid of this structure. [7, 8] Mitchell et. al. [7] improved this wavefront propagation method and named it as “continuous Dijkstra”. In “continuous Dijkstra”, a “signal” is propagated from source to the rest of the surface. Each time a point records the shortest distance from source when it receives this “signal” and propagate it further. The pre-processing operation can be done in time $O(n^2 \log n)$, then the shortest path is reported in time $O(k + \log n)$, where k is the number of faces crossed by the path and n is the total number of vertices. Later on, Chen & Han [2, 3] proposed an different algorithm other than “continuous Dijkstra” technique. With this algorithm, the shortest path between two vertices can be reported in time $O(n^2)$ without pre-processing the surface into subdivision. A sequence tree structure is built during the process of unfolding all the faces of the polyhedral surface. Each node of the tree represents a set of shortest path which all have the same edge sequence and angularly contiguous at the source. Chen & Han avoid the exponential trap by keeping the “one single one split” property throughout tree structure construction. Chen & Han algorithm is currently one of the fastest and only feasible method to search the exact shortest path on polyhedral surface. Both of them could deal with the non-convex polyhedral surface. Clearly, the surface partitioning facilitates the

quick report of the shortest path. But it does not favor all cases. For example, in the battle surveillance system and wildlife behavior monitoring system, only few objects are our real interests. And the objects including source itself are always in moving situation. In such circumstance, the costly surface subdivision for moving source point will not be an attractive choice.

With the attempt to balance the cost in time and approximation accuracy, various methods are developed to find approximate shortest path [4, 10–13]. Takashi et. al. [4] proposed an algorithm for single-pair problem on a polyhedral surface, which mainly used Dijkstra’s algorithm and is based on selective refinement of the discrete graph of the polyhedron. In their method, Dijkstra’s algorithm is iteratively used to narrow down the region in which the shortest path can exist. This approximate algorithm can calculate shortest path within 0.4% accuracy to roughly 100 – 1000 times faster comparing with Chen & Han algorithm in experiment. Away from the approximate algorithms, Dinesh et. al. [6] developed an approach to solving the single pair problem for the robotic system motion planning. Their approach computes a multiresolution representation of the terrain using wavelets, and hierarchically plans the path through sections, which are well approximated on coarser levels and relatively smooth. These two approaches described above concentrate on the single pair shortest path problem and they try to narrow down the search area and thus simplify the problem.

However, all these algorithms mentioned in this section need to traverse the entire surface to find the single pair shortest path. Moreover, the approximate method cannot report exact shortest path. Dinesh’s algorithm does not offer a comprehensive experiment result about their algorithm’s performance. In contrast, one contribution of our algorithms in this paper is that our approach does not require navigating the entire polyhedral surface to solve single pair shortest path problem. The essential method we used is to select the search region in a carefully designed range and test the local shortest path using the global optimal criteria. The shortest path search terminates once the global optimal criteria are satisfied.

3 Algorithms

In this section, three approaches to obtain the shortest path on the polyhedral terrain surface are detailed, where surface is represented as meshes consisting of triangular planar. Let $V = \{v_i | i = 1 \dots n\}$ be a set of vertices on the polyhedral terrain surface G^1 , where v_i denotes each from s to d is represented by p_{sd} . The xy -plane in three dimensional coordinate system is defined as $Plane_{xy}$. In this paper, because we work on the polyhedral terrain surface, only xy -plane will be discussed. We also denote the straight line between points s and point d as sd and the orthogonal projections of s and d in xy -plane are s' and d' , there is

¹ A polyhedral terrain is a polyhedral surface that every vertical line intersects it in at most one point. The polyhedral terrain surface may be convex or non-convex, with or without boundary.

following relation $s'd' = sd \times \cos\Theta$. Here, Θ is the angle between sd and $s'd'$. In geometry, Euclidean distance corresponding to the straight line between two points is the shortest, that is, any path connecting two points on the surface are always longer than or equal to the line segment of their projection on coordinate plane. This concept is applied in the rest of the paper.

To compute the shortest path more efficiently, we apply geometric rules on polyhedral terrain surface to identify the maximum possible region, in which the global shortest path exists, and can only exist within this region. Our algorithms show that any path going beyond the border of the selected region will not be the shortest one.

We develop three algorithms in this paper: Naïve algorithm, RB and EB to find the global optimal shortest path. Each algorithm contains two steps: (1) compute the initial path connecting s and d . This step should commit two objectives, computing at low cost and returning optimum initial path. However, optimum initial path is achieved at the expense of computing cost, and vice versa. we design the first step to obtain the initial path at low cost in Naïve and RB algorithm. Whereas in EB algorithm, we design this step by considering the trade-off between those two objectives. (2) select the region that encloses the global shortest path. We offer four methods to select the region as will be shown in the following subsections. Technically, the range of the selected region is determined by the length of the initial path. Once we select the search region, we use Chen & Han algorithm to search the shortest path. Currently their algorithm is the only simple and feasible method of computing the shortest path, yet other algorithms could also be applied. When the shortest path is found in the selected region, the global optimum criteria must be tested to certify its optimum in global. Global optimum criteria are a set of rules which is closely related with the methods in each algorithm. If the global optimum criteria are satisfied, the current shortest path is also global optimum. Otherwise, the shortest path will be search again in the larger area until the criteria are satisfied.

3.1 Naïve Algorithm

Step One. In this algorithm, The method used to search the initial path is very fast. The idea is that any path connecting s and d on the surface can be the initial path. For the sake of simplicity, we choose one path whose projection in xy -plane is the line segment $s'd'$, see Fig. 1 (a). All the triangles that the initial path p_{sd} passes through form a triangle sequence. By computing every component of the initial path in the triangle sequence, the length of the initial path is the sum of these segments. This step is done in time $O(n)$, where n is the number of triangles in the minimum bounding rectangle (MBR). In Fig. 1 (b), the gray rectangle is the MBR in xy -plane where s' and d' are its vertices. Clearly, the initial path P_{sd} calculated by this way is not local optimum, because it can not satisfy the requirement of the shortest path. Generally, the shortest path should be built on local optimum that the path enter and leave the edge in the same angle, and it is a straight line after unfolding each triangle planar.

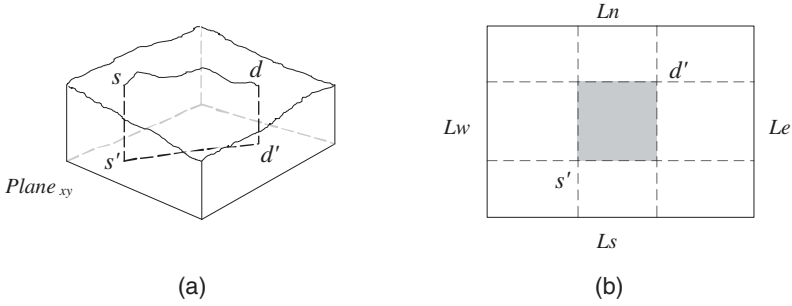


Fig. 1. Naïve algorithm (a) initial path estimation (b) Selected region with new borders.

Step Two. Once we calculate the initial path, we can select the search region by extending from the MBR used for initial path. In each of the four directions, a new border L_w , L_e , L_n and L_s should be identified, as being depicted in Fig. 1 (b). The sum of the vertical distance from point s' to line L_w and the vertical distance from point d' to line L_w should be equal to the length of the initial path, so do other three directions. In this case, any path connecting s' and d' in xy -plane is longer than the initial path if it goes beyond the new selected region bounded by solid lines. In other words, any path connecting s and d must be inside the new selected region. passing through any point whose projection is outside the new selected border, the path is longer than the initial path.

This algorithm uses an easy way to search the initial path. The advantage is its low cost. However, because the initial path is in a pre-determined direction along the surface, it may be happen to pass the roughest terrain region. The length of the initial path may turn out to be severely skewed. In turn, the long initial path will lead to an unnecessary large search region in step 2. In some cases, the cost saved in initial step will be consumed by the shortest path search computation in step 2.

3.2 Rectangle Border Algorithm

We offer two alternatives under this algorithm in terms of the way of testing global optimum criteria and extension methods, namely RBD and RBID. Both of them using the same approach in step one but differ in step two.

Step One. Due to the inherent weakness of the initial path searching method in Naïve algorithm, the search region in step two may be unnecessarily large. By finding the local optimum initial path, this problem can be avoided. We draw a rectangle region around points s' and d' . This rectangle is a bit wider than the MBR in Naïve algorithm. The surface bounded by this rectangle are used to search the local shortest path connecting s and d . Any point on the bounded surface has its projection within the rectangle in xy -plane. If we use Chen & Han's algorithm as the basic shortest path search algorithm, the local shortest

path is found in time complexity $O(n^2)$, where n is the number of triangles in the search region. Even though the cost in this step is much higher than $O(n)$ in Naïve algorithm, this method has its own advantages: (1) the output initial path is local optimal that could optimize the search region in step two. (2) the current local shortest path may also be the global optimum, yet we need to apply the global optimal criteria for this.

Step Two

Rectangle Border Indirect Algorithm: The local optimum initial path found in step one may or may not be the global optimum, therefore we first apply global optimal criteria test against the output. In this method, the global optimum criteria make use of the rectangle border as a special component of the surface. During the searching process, the path can traverse along the surface, or if the path reaches the border, it can take the shortcut along the border.

Chen & Han's algorithm can directly report the shortest path between vertices. However, it can not easily find shortest path for any point inside the triangle on the polyhedral surface. If the path from the source point to any point inside the triangle is required, this point needs to be triangulated as a vertex. To simplify the problem, the distance to any point inside a triangle can be defined in a certain range. This is the Lemma 1.

Lemma 1. *Let e_1, e_2 and e_3 be the edges of a triangle on the surface, and p_1, p_2 and p_3 be the shortest paths from the source point s to the three vertices v_1, v_2, v_3 . If $e_1 > e_2 > e_3$ and $p_1 > p_2 > p_3$, the shortest path p_r from s to any point r inside the triangle will be $p_1 + e_1 \geq p_r \geq p_3 - e_1$*

Proof. If any point inside this triangle does not follow above relations, its shortest path p_r is less than $p_3 - e_1$, then we have the relation $p_r + e_1 < p_3$. Because point r and three vertices are on the same planar, and e_1 is the longest edge, the distance $dist_r$ from point r to the vertices must be shorter than e_1 . That is $dist_r < e_1$. So $p_r + dist_r < p_r + e_1 < p_3$. This indicates that there is a path $p_r + dist_r$, which is less than the known shortest path p_3 . This contradicts the fact that p_3 is the shortest path. Therefore, $p_1 + e_1 > p_r$ can be proved. \square

The rectangle border is cut into line segments by the triangles whose projections in xy -plane intersect the border. Then, the border can be treated as line segments instead of points. For example, in Fig. 2, ab is one of the line segments. If all border-touched path is longer than or equal to the local shortest path, the local shortest path must be the global optimum. In this case, global optimum criteria are satisfied.

Theorem 1. *Given a triangulated surface G , the local shortest path connecting points s and d inside a rectangle region is represented as p_{sd} . Let p_1 be the shortest path from s to line segment L_1 and p_2 be the shortest path from s to line segment L_2 . See Fig. 3. The path along the border connecting the two line segments is p_{12} , which value is the shortest in space. If for all pairs of border segments: $p_{sd} \leq p_{12} + p_1 + p_2$, the global optimum criteria are satisfied, that is, p_{sd} is the global shortest path. \square*

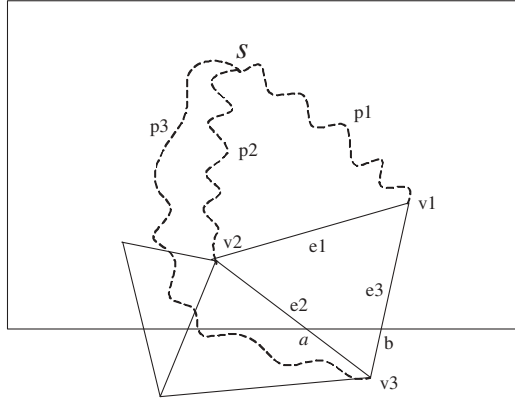


Fig. 2. RBID shortest path range for non-vertex points.

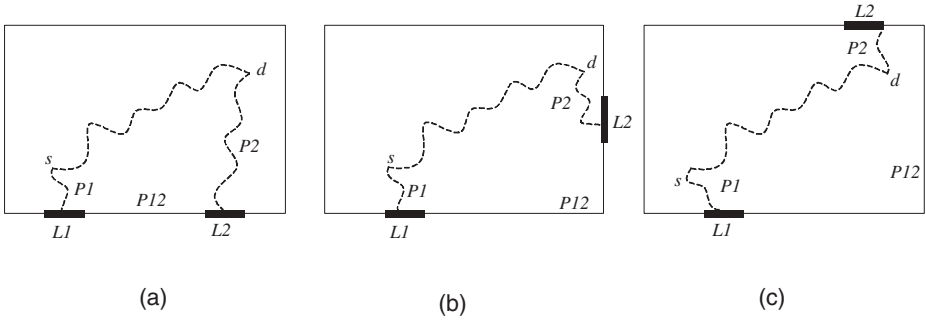


Fig. 3. Three situations of global optimum criteria.

Because all the border line segments must satisfied the $p_{sd} \leq p_{12} + p_1 + p_2$, there are three situations need to be considered, see Fig. 3. According to Lemma 1, the range of p_1 and p_2 can be estimated with the length of the local optimum shortest paths to the three vertices. The path connecting L_1 and L_2 along the border is p_{12} and it must be the shortest in space. In Fig. 3 (a), p_{12} is a straight line segment on the border and it is the shortest. In Fig. 3 (b) and (c), the border is not a straight line. If p_{12} takes the value of the border length, it cannot guarantee that p_{12} is the shortest in space. In such situation, we use the Euclidean distance to replace p_{12} . If $p_{sd} > p_{12} + p_1 + p_2$ for some line segments, the global optimum criteria are not satisfied. In this situation, the search region must be extended to obtain more surface information, then search the shortest path in this extended region again. One point to note that, because p_1 and p_2 have difference source s and d , we need to search the region twice to find the shortest path.

In Naïve algorithm, the method used to extend the search region gives a too wide extension range. We offer a better extension strategy in this RBID algorithm. When testing the border condition for the global optimum criteria,

we have known which pair of border line segments do not satisfy the criteria and to what extent the extension should be. The difference between p_{sd} and $p_{12} + p_1 + p_2$ (see Fig. 3) sets the upper bound for range of extension. So, ellipse can be used. In geometry, an ellipse is a set of points in a plane the sum of whose distances from two foci is a constant (we call this constant as ellipse constant in the rest of the paper). For points outside the ellipse, the sum of distance to both foci is greater than the ellipse constant. This feature can be employed to identify better extension range. In Fig. 4, points a' and b' are on the border of the initial rectangle which are the orthogonal projection of points a and b . In xy -plane, an ellipse is drawn with a' and b' as the foci and the ellipse constant is equal to the difference between p_{sd} and $p_{12} + p_1 + p_2$. r is a point on the surface and r' is the projection of r . According to the nature of ellipse, $a'r' + b'r'$ is the ellipse constant if r' is happen to be on the ellipse. Obviously, at the same time, the sum of the shortest path p_{ar} and p_{br} on the surface is longer than or equal to the ellipse constant. On the other side, if r' is outside the ellipse, $p_{ar} + p_{br}$ must be greater than the constant. That is, if r' is outside the ellipse, $p_{sd} < p_{sa} + p_{ar} + p_{rb} + p_{bd}$. So, the ellipse-bounded region is the region that may contain shorter path than initial path. For simplicity, the tangent line of the ellipse is used as the new border for this border line segment pair, as depicted in Fig. 4.

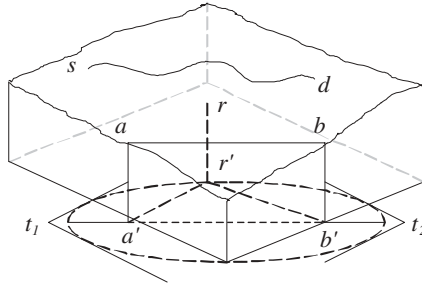


Fig. 4. RBID extension strategy.

Note that the extension approach above is only for border line segments containing a' and b' . To decide the extension range, all the border line segment pairs with $p_{sd} > p_{12} + p_1 + p_2$ should be considered. Finally the biggest extension value will be chosen as the extension range.

Rectangle Border Direct Algorithm : Besides the global optimum criteria testing and extension method discussed above, an alternative approach can be used. After the local shortest path is found, an ellipse is drawn with s' and d' as foci and ellipse constant is equal to the length of initial local shortest path. Using this method, we avoid checking all border line segment pairs. This is a straightforward method to test the global optimum criteria. If the range of ellipse is inside the initial rectangle, the global optimum criteria are satisfied. If ellipse is beyond the initial rectangle border, we need to search the shortest path again with the new

region defined by the ellipse. Obviously, this method is easy to implement and get even better performance. because it does not require searching the selected region twice.

3.3 Ellipse Border Algorithm

Step One. The approach used in step one of Naïve algorithm can quickly find an initial path, but it is not local optimum. On the contrary, RB algorithm can obtain the local optimum initial path, however with high resource cost. Both of them have advantages and disadvantages. In this section, a more efficient approach is developed to search the initial path by taking into account of surface roughness.

Our approach estimates the initial path p_{sd} using the following heuristics, see Fig. 5 (a). First, the initial path p_{proj} is computed using the same method as in step one of Naïve algorithm. The orthogonal projection of p_{proj} is the line segment $s'd'$, which is the projections of s and d in xy -plane. Second, an ellipse with s' and d' as its foci is drawn and the ellipse constant equals to p_{proj} . The surface inside the ellipse is equally partitioned along the direction from s to d . The average height $h_{average}$ of each part is computed. Then, the path $p_{average}$ along the surface is computed. Next, we compare p_{proj} and $p_{average}$. If p_{proj} is close to $p_{average}$, the estimate value of initial path will be the length of p_{proj} . The closeness is indicated by ratio $\delta = \frac{p_{average}}{p_{proj}}$. If $\delta > 1.2$, the estimate value of initial path will be the length of $p_{average}$. This implies that p_{proj} passes through a very rough area.

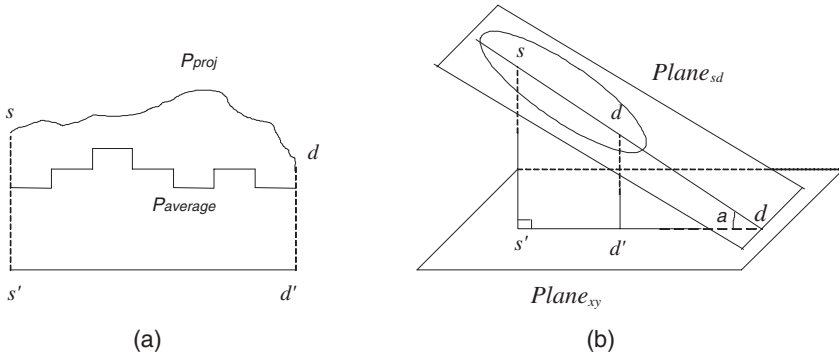


Fig. 5. Ellipse algorithm (a) Initial path estimation (b) Ellipse search region selection.

Step Two. After the initial path is found, the search area needs to be selected. We draw an ellipse in sd -plane with the initial path as the ellipse constant, and let s, d as ellipse's foci rather than their projections s' and d' , see Fig. 5 (b). The sd -plane is the plane that contains s and d , and intersects xy -plane at angle ω . This choice can further decrease the search region. We let a equal to the value

of the initial path, c equal to half of the Euclidean distance between s and d , b as the width of the ellipse. There is following relation:

$$b = \sqrt{a^2 - c^2} . \quad (1)$$

Clearly, b value declines as c value increases. Because the ellipse is drawn in sd -plane, c has greater value than that of in xy -plane, therefore, search region selected in sd -plane is smaller than that of in xy -plane. After the search region is identified, we can compute the shortest path in this area and this shortest path is the global optimum. Following discussion provides the proof.

Given a local shortest path p_{short} connecting points s and d inside an ellipse area on the surface, if and only if the length of the local shortest path p_{short} is shorter or equal to the ellipse constant, the global optimum criteria are satisfied. That means p_{short} is the global shortest path. This can be easily proofed. Any path p_{sd} connecting source point s and destination point d traverses beyond the ellipse border is longer than the ellipse constant Q , $p_{sd} > Q$. If the local shortest path $p_{short} \leq Q$, it is obvious that $p_{short} < Q < p_{sd}$. The global optimum criteria is satisfied and p_{short} is the global shortest path.

In our case, the ellipse is drawn with length of initial path as the ellipse constant Q . Because the initial path must be longer than or equal to the local shortest path, thus $p_{short} \leq \text{initial path}$. As $\text{initial path} = Q$, therefore, $p_{short} \leq Q$. We can conclude the output local shortest path must also be the global optimum.

4 Performance Study

In this section we compare the three algorithms discussed in this paper. The primary performance index used in this section is the processing time, which is measured as the elapsed time from the point of parameters (start and destination points) are set to the global shortest path is reported. As we employed a two-step approach in the proposed algorithms, we test not only the whole processing time, but also the time spend in each step. According to Kaneva and O'Rourke's experimental report [1], the I/O cost-based measure such as the number of disk pages accessed is not an significant performance indicator because Chen & Han algorithm is typically CPU intensive, thus a small amount of difference in the input data leads the computing time to a superlinear growth. Therefore I/O cost is not measured in our analysis.

4.1 Cost Analysis

The time used to search the global shortest path, T_{total} , is the sum of time used in step one T_{s1} and step two T_{s2} . That is:

$$T_{total} = T_{s1} + T_{s2} . \quad (2)$$

Different methods used in step one of each algorithm are in different time complexity that determine T_{s1} . Therefore, in Naïve and EB algorithms, step one has

time complexity $O(n)$, whereas in RB algorithm, step one has time complexity $O(n^2)$. In step two, the time T_{s2} is decided by the time complexity of which the algorithm is chosen. As we choose Chen & Han algorithm as the basic shortest path algorithm in the analysis and experiment, T_{s2} is in time complexity $O(n^2)$. It is clear that our algorithms do not change time complexity of Chen & Han algorithm. Thus the time complexity for all of these three algorithms should be $O(n^2)$.

We use the projection of the selected region in xy -plane, $Area_{s1}$ and $Area_{s2}$ to represent the search region in two steps respectively. In the initial step, the number of triangles in $Area_{s1}$ is represented as n_1 . Similarly, the number of triangles in $Area_{s2}$ is represented as n_2 for step two. Because $Area_{s2}$ is extended from $Area_{s1}^2$, n_2 is normally larger than or equals to n_1 , therefore, the processing time is mainly decided by n_2 . Given source point s and destination point d , n_2 is determined by $Area_{s2}$.

Let a be the half length of the initial path, c be the half of the distance of sd and c' be the distance of $s'd'$. The area for selected search region in step two of Naïve algorithm can be estimated by:

$$Area_{s2} = 4a^2 . \quad (3)$$

For RB1 algorithm, the area of the selected search region is:

$$Area_{s2} = 4a\sqrt{a^2 - c^2} . \quad (4)$$

For EB algorithm, the area of the selected search region is:

$$Area_{s2} = 4a\sqrt{a^2 - c^2} . \quad (5)$$

c is the projection of c' in xy -plane, $c = c' \times \cos\beta$. β is the angle between line sd and xy -plane. Theoretically, for a given length of initial path, EB algorithm will get the optimum selected search region, followed by RB algorithm and the Naïve algorithm. However, if the length of initial paths are not the same, the total performance for each algorithm will differ.

Three factors influence the number of triangles in a given region, and consequently affect the processing time. The first factor is the resolution of the triangulated surface. Surface with higher resolution contains more triangles, thus n is larger for a given area. It has different impacts on T_{s1} of three algorithms but same for the second step. T_{s1} for $O(n^2)$ RB algorithm increases in a quadratic manner, while for $O(n)$ EB and Naïve algorithm T_{s1} increases in a linear manner. The second factor is the roughness of the surface which affects the number of triangles as well. The influence of the roughness on processing time follows the same pattern of the first factor. Third factor is the angle φ between the line crossing the s' and d' in xy -plane and the x axis. Because the projection of the surface region used in initial step is a rectangle which diagonal vertices are s' and d' . And the sides of rectangle is parallel with the x , y axis respectively. The relation between φ and $Area_{s1}$ is:

$$Area_{s1} = \frac{1}{2} \sin(2\varphi) dist_{s'd'} . \quad (6)$$

where $dist_{s'd'}$ is the Euclidean distance between s' and d' . The formula shows $Area_{s1}$ is the maximum when φ is $\frac{\pi}{4}$. Angle φ affects step one of all algorithms, yet step two of EB, RBD and Naïve algorithm is independent of this factor. Because of the obvious effect of first two factors on performance, we conduct the experiment concentrating on the last factor.

4.2 Experimental Results

In this section, we evaluate the performance of three proposed algorithms for finding the shortest path, namely Naïve, RB and ER algorithm. Two extension methods are tested separately for RB algorithm: RBD and RBID. Both development and testing are done using PC with Pentium IV 2.8G CPU and 512M memory. The operating system is Windows XP Professional. Three algorithms are implemented with C++ compiled with Borland C++ 5.5 compiler. The Microsoft ODBC library is used in the C++ program in order to access the data set. We use a real polyhedral terrain surface data sets in the test, provided by MinCom Pty. Ltd. This data set consists of 10806 triangles and is stored in Oracle Enterprise Edition 9.2. Indexes are created wherever necessary for all the tables. In this experiment, we use the Chen & Han algorithm implemented by Kaneva and O'Rourke [1] to search the shortest path. The performance of proposed algorithms is tested in terms of the processing time.

We first test the effectiveness of proposed Naïve, RBD, RBID and EB algorithms in reducing the time cost. The test is conducted by varying $dist_{s'd'}$ (ranging from 100 to 500, the unit is the same as the coordinate value of the triangles) which is the Euclidean distance between s' and d' . At each $dist_{s'd'}$, we randomly select ten pairs of source and destination points on the surface. The time usage of the sample data presented is the mean value. Figure 6 (a) shows the time comparison between Naïve algorithm and Chen & Han algorithm. In order to find the global optimum shortest path, Chen & Han algorithm must compute the entire surface. Instead we select the nonconvex surface with 5175 triangles. Since time spend to report the single pair shortest path by using their algorithm is over 10,000 seconds and is indifferent to the distance changes, we could set this time as a lower time bound for the comparison purpose. Obviously, our proposed algorithm results in a substantial time reduction, i.e. time spend at 500 distance units is less than 500 seconds. Figure 6 (b) illustrates the different time cost among our proposed algorithms with varying $dist_{s'd'}$. As we can see from the result, the total time used in Naïve and RBID algorithms are very close and increase drastically as the distance becomes larger. Not surprisingly, RBD and EB algorithms outperform the other two. The reason for this difference is explained in Fig. 7.

In Fig. 7, the time cost in each step is drawn separately. Consistent with our cost analysis, where the time complexity in step one for RBID and RBD is $O(n^2)$, for EB and Naïve algorithm is $O(n)$. Figure 7 (a) shows the time difference among these four algorithms. Both EB and Naïve algorithms have similar time cost as they are base on the same time complexity. One point to note that time spend in RBID is doubled comparing with RBD even though they have the same time

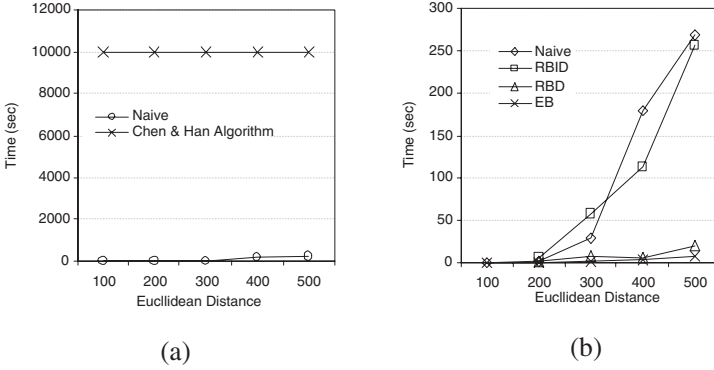


Fig. 6. (a) Naive vs. Chen & Han algorithm (b) comparison between proposed algorithms.

complexity. Because in order to test the global optimum criteria, RBID needs to search the selected region twice. Figure 7 (b) reveals an important fact that step two is the dominant factor in the total time cost. The step one has little effect because of its small proportion to the overall time cost. Evidence shows that extension method plays an essential role in the total time cost. Extension methods proposed in EB and RBD are superior to others, therefore, these two algorithms have an outstanding performance. These results strongly support our cost analysis above.

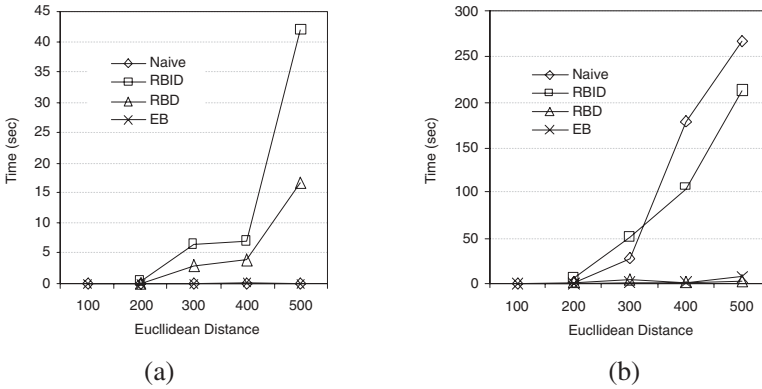


Fig. 7. (a) Time cost in step one (b) Time cost in step two.

Figure 8 depicts the influence of angle φ on the processing time for all approaches. We randomly select ten pairs of source and destination points for a certain degree, i.e. $15^\circ, 30^\circ, 45^\circ$ etc. As we cannot obtain the ten pairs of data at the exact same degree, we take the average degree of each group of data. The angle range starts at 0 and ends at $\frac{\pi}{2}$. As we discussed in previous section, the angle influences all approaches in step one but rather moderate, and has no

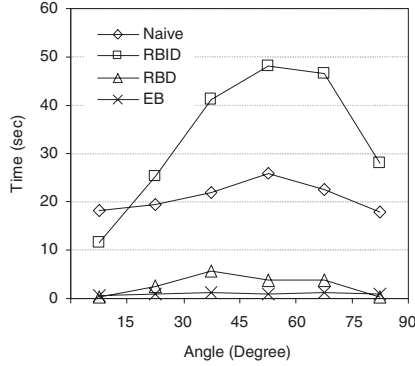


Fig. 8. The relation between angle φ and processing time.

effect on Naïve, RBD and EB algorithms in step two except RBID because the methods used to select search region for these three approaches are free of the φ change. It has the maximum impact when $\varphi = \frac{\pi}{2}$. The results shown in Fig. 8 support our analysis.

The above discussions demonstrate that all proposed algorithms can significantly improve the performance of the basic shortest path search algorithms. Among them, EB and RBD approaches are more efficient comparing to RBID the Naïve algorithms in terms of time usage, especially in longer distance.

5 Conclusions

With rapid advances in technology, the surface models are becoming more and more complex. The single pair shortest path problem is getting harder and harder to solve. This paper proposed three expansion-based algorithms, namely Naïve algorithm, RB and EB. We also offered two alternatives under RB algorithm in terms of the way of testing global optimum criteria and extension methods: RBD and RBID. In contrast to current algorithms which need to traverse the whole surface to search the global shortest path between a pair of points. This operation is very time-consuming. Motivated by the observation that the single pair shortest path is in the locality, we developed our algorithms based on a two-step strategy: (1) compute an initial local path. (2) use the value of this initial path to select a search region, in which the global shortest path exists. The search process terminates once the global optimum criteria are satisfied. By reducing the searching region, the performance is improved dramatically in most cases. We have shown a comparison between our algorithms and Chen & Han algorithm by using a real data set. Experimental results shows that all of the proposed algorithms can significantly improve the performance of Chen & Han's algorithm, whereas the EB and RBD notably outperform RBID the Naïve algorithm in term of time usage, especially in longer distance. In the future, we plan to integrate the advanced multiresolution technology into our shortest path algorithms.

Acknowledgments

The work reported in this paper has been supported by an Australian Research Council Discovery Project grant (grant number: DP0345710). We also would like to thank Mincom Pty Ltd for providing access to their terrain data and thank Prof. Hong Shen, Dr. Kai Xu for many helpful discussions.

References

1. B. Kaneva, J. O'Rourke: An Implementation of Chen & Han's Shortest Paths Algorithm. In *Proc. of the 12th Canadian Conf. on Comput. Geom.*, pages 139-146, 2000.
2. J. Chen, Y. Han. Shortest paths on a polyhedron. In *6th ACM Symposium on Computational Geometry*, Pages 360-369, 1990.
3. J. Chen, Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127-144, 1996.
4. T. Kanai, H. Suzuki. Approximate shortest path on polyhedral surface based on selective refinement of the discrete graph and its applications. In *Geometric Modelling and Processing*, pages 241-250, 2000.
5. T. Deschamps, L. D. Cohen. Minimal Paths in 3D Images and Application to Virtual Endoscopy. In *Proc. sixth European Conference on Computer Vision (ECCV'00)*, Dublin, Ireland, 26th June - 1st July 2000.
6. D. K. Pai, L. -M. Reissell. Multiresolution Rough Terrain Motion Planning. *IEEE Transactions on Robotics and Automation*, 14 (1): 19-33, February 1998.
7. J. S. B. Mitchell, D. M. Mount, C. H. Papadimitriou. The Discrete Geodesic Problem. *SIAM J. Comput.*, 16: 647-668, 1987.
8. M. Sharr, A. Schorr. On Shortest Paths in Polyhedral Space. *SIAM J. Comput.*, 16: 647-668, 1987.
9. C. S. Mata, J. S. B. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. In *Proc. 13th ACM Symp. On Computational Geometry*, pages 265- 273, 1997.
10. M. Lanthier, A. Maheshwari, and J.-R. Sack.. Approximating weighted shortest paths on polyhedral surfaces. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 274-283, 1997.
11. K. R. Varadarajan and P. Agarwal. Approximating shortest paths on a nonconvex polyhedron. In *Proc. 38th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 182-191, Miami Beach, Florida, 20-22 October 1997.
12. S. Har-Peled. Constructing approximate shortest path maps in three dimensions. *SIAM J. Comput.*, 28:1128-1197, 1999.
13. P. K. Agarwal, S. Har-Peled, M. Sharir, and K. R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. ACM*, 44:567-584, 1997.

MR-Tree: A Cache-Conscious Main Memory Spatial Index Structure for Mobile GIS*

Kyung-Chang Kim and Suk-Woo Yun

Dept. of Computer Engineering, Hongik University, Seoul, Korea
{kckim, sw0305}@cs.hongik.ac.kr

Abstract. As random access memory chips get cheaper, it becomes affordable to realize main memory-based database systems. The most important issues in main memory database indexing techniques for mobile GIS applications are to improve update performances -since numerous update operations can arise for tracking continuously moving objects- and to reduce cache misses for improving search performances. In this paper, we propose MR-tree, a cache-conscious version of the R-tree for main memory databases. To increase fan-out, the MR-tree applies a novel compression scheme to entry MBRs (Minimum Bounding Rectangles). This scheme represents entry MBRs by relative coordinates in a node. To improve update performance, the MR-tree can become an unbalanced tree as follows: it propagates node splits upward only if one of the internal nodes on the insertion path has empty space, or all height differences between the consecutive nodes on the insertion path are 1 and one of the height differences among its subtrees is equal to some unbalance parameter. Because of this feature, the MR-tree can reduce the number of internal nodes splits and reinsertions significantly. In the case where all internal nodes on the insertion path do not meet the above conditions, a newly created leaf node simply becomes the child of the split leaf, when a leaf node split occurs. This split leaf is called the half-leaf node. Our experimental result shows that the search speed of the proposed two-dimensional MR-tree increases by almost a factor of two compared to the CR-tree and the R-tree variant, which are also main memory based R-trees, while maintaining better update performance.

1 Introduction

The most significant problems of all mobile GIS applications - including traffic control or monitoring, transportation and supply chain managements, digital battlefields, and mobile e-commerce - are to rapidly track the current positions of continuously moving mobile users while supporting almost real-time responses during data access [11]. Main memory-based database systems [10] can be used to solve these problems. However, adapting traditional spatial index structures such as the R-tree [7] variants to main memory database systems is not suitable for tracking these positions, on account of numerous update operations.

To support almost real-time responses in main memory environments, search performances must also be improved. The search time of spatial index structures consists

* This work is supported by KOSEF grant # R01-2001-000-00540-0 (2003).

of the node accessing time and the key comparing time. Today, the increase in CPU speeds at 60% per year and memory speeds at 10% per year makes it possible to reduce the overhead of comparing index key entries more and more [6]. In one study [4], the authors reach the conclusion that level 2 cache misses occupies a significant portion of execution time. A cache miss occurs when the requested data is in main memory but not in cache. At this point, reducing cache misses is the most important factor that affects the search performance in indexing techniques for main memory databases [3].

Using the R-tree [7] variants in main memory for the multidimensional index structure, or a pointer elimination technique of the CSB⁺-tree [6][5] alone cannot widen the index tree significantly, since multidimensional keys called MBRs (minimum bounding rectangles), are much larger than pointers [1].

In this paper, we propose a modified cache-conscious version of the R-tree, which we call an MR-tree. For our purposes, the term cache-conscious means to reduce cache misses. To be cache-conscious, the MR-tree uses a new compression scheme applied to MBRs. In this scheme, an MBR is represented relative to the lower bounding coordinates of the node MBR, and the size of this relative MBR varies according to index nodes. It is not sufficient to apply the MR-tree to the mobile GIS applications in which objects to be indexed change positions frequently, because the MR-tree can cause a lot of node splits and reinsertions when tracking moving objects. We, therefore, add another feature in that we permit the MR-tree to be unbalanced. In the MR-tree, height differences among the nodes that have the same parent are equal to or less than the unbalance parameter, *diff*. This way, we believe that the MR-tree can reduce the number of node splits and reinsertions considerably. The experimental result shows that the MR-tree outperforms CR-tree and R-tree significantly in terms of the search and update performance.

We organize this paper as follows. In Section 2, we discuss the related works briefly. In Section 3, we explain the basic idea of an MR-tree, followed by the details of its index structure, search, insertion, and deletion algorithms, and in Section 4, we present the results of various experiments conducted to test the performance of the MR-tree. A conclusion is provided in Section 5.

2 Related Work

In recent years, many spatial index structures have been proposed as shown in [13]. Among index structures, Kothuri et al. [14] argue that R-trees are generally better than quad-trees and Oracle now recommends the use of only the R-tree. Our approach is also based on the R-tree.

Some new index structures have been recently proposed for indexing moving objects. These index structures can be classified into (1) trajectories (histories) and (2) current positions of objects. Our approach belongs to the latter category.

Most index structures for moving objects only focus on update performance. But in the real world, most moving objects - such as mobile users - are in an almost static state most of the time [12] while requiring near real-time responses. We therefore focus on the search performance as well as the update performance. In this section, we briefly survey the two index structures that inspire our approach.

The CR-tree [1] is a cache-conscious version of the R-tree. To pack more entries into a node, the CR-tree uses a lossy compression scheme called QRMBR. The index node of the CR-tree consists of one reference MBR and multiple entries, which are multiple QRMBR keys and pointers to child nodes. The reference MBR represents the MBR of the index node. Let L , R and E be the quantization level, reference MBR and entry MBR respectively, and then the lower bound xl and upper bound xu of QRMBR Q are defined as follows:

$$Q.xl = \begin{cases} 0, & \text{if } E.xl \leq R.xl \\ L-1, & \text{if } E.xl \geq R.xu \\ \lfloor L(E.xl - R.xl)/(R.xu - R.xl) \rfloor, & \text{else} \end{cases} \quad Q.xu = \begin{cases} 1, & \text{if } E.xu \leq R.xl \\ L, & \text{if } E.xu \geq R.xu \\ \lceil L(E.xu - R.xl)/(R.xu - R.xl) \rceil, & \text{else} \end{cases}$$

The CR-tree has two significant problems in adopting it as an index structure for mobile applications. First it causes a lot of node splits and reinsertions, like the R-tree. Second, it deteriorates the search performances, since the lossy compression scheme can cause anomalies, i.e., it can choose a wrong insertion path because QRMBRs are enlarged falsely whenever recalculated. This behavior increases the number of cache misses unnecessarily during search, because the number of nodes to visit increases significantly. Figure 1 shows an example.

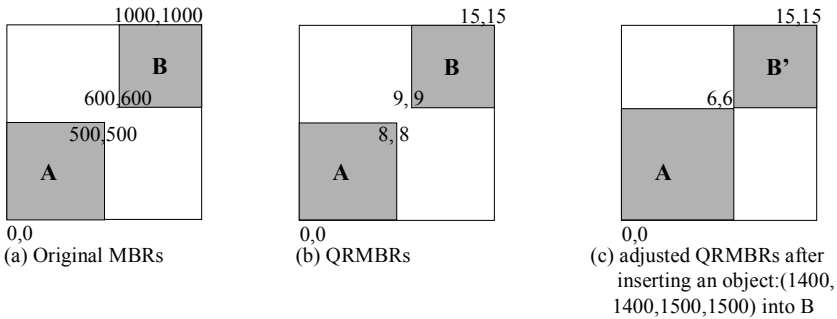


Fig. 1. An example of the QRMBR anomaly (Quantization level = 4 bits).

In figure 1, original MBRs of an internal node are represented in (a), and QRMBRs in (b). When a new object having (1400,1400,1500,1500) is inserted to the CR-tree, it chooses QRMBR B for the insertion path, and recalculates QRMBRs after inserting into a leaf. Figure 1(c) shows this result. From figure 1(c), we know that QRMBR A has been falsely increased, because its correct value is (0,0,5,5). When another object having (550,550,600,600) is inserted into the CR-tree, it chooses A, but the right insertion path is to choose B'.

The LUR-tree [2] is also an R-tree variant that uses the Lazy Update approach to reduce the number of update operations. This approach updates the structure of the index only when an object moves out of the corresponding MBR. If the new position is still within the MBR, only the position of the object in the leaf node is updated. The approach shows fairly good update performances for indexing mobile GIS applications, in which an object is moving within a small region of space. This algorithm, however, is of no use when an object moves out of the region of corresponding node.

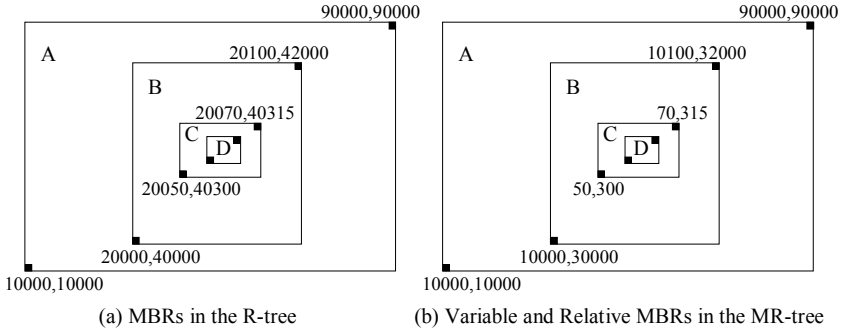


Fig. 2. Variable and Relative representation of MBR.

3 MR-Tree

3.1 Basic Idea

Because of anomalies as mentioned in Section 2, the CR-tree shows bad search performances when the approximation level is too small. The idea in this paper, therefore, is to make R-tree cache-conscious by applying different compression levels to index nodes. Figure 2(a) shows the absolute coordinates of A~D. In this case, the sizes of all entry MBRs in nodes of R-tree use at least 68 bits because the R-tree needs 17 bits ($=\lfloor \log_2 90000 \rfloor + 1$) to represent maximum coordinate value. As compared to the R-tree, however, we reduce the size of entry MBRs by using variable and relative MBRs, called VRMBRs, the sizes of which can vary according to index nodes. VRMBR is an entry MBR in a node and a relative representation to minimum bounding coordinates of a node MBR. For the sake of simplicity, we use the same size of VRMBRs in the same node, but the sizes of the VRMBRs can vary with index nodes. In figure 2(b), entries of node B use 44-bit VRMBRs, since they need 11 ($=\lfloor \log_2 (32000 - 30000 + 1) \rfloor + 1$) bits to represent a coordinate in node B. Entries of node C, however, need 5 ($=\lfloor \log_2 (70 - 50 + 1) \rfloor + 1$) bits for their coordinates, thus sizes of VRMBRs of node C are only 20 bits. From this example, we know that VRMBRs in low levels are smaller than those in high levels. Fanouts of nodes, therefore, become large when they are closer to the leaf level resulting in increase in search speed. Our proposed index structure, called the MR-tree, is a cache-conscious R-tree, since it reduces the number of cache misses by using VRMBRs as index keys.

To adopt the MR-tree as an index structure of mobile GIS applications, we use the Lazy Update [2] approach. The MR-tree, however, needs additional schemes to increase insertion and deletion performances, since the Lazy Update approach is of no use when an object is out of the region of the corresponding node. We, therefore, propose that the MR-tree can be unbalanced to reduce the number of node splits and reinsertions. Details of the “unbalance” are explained in following sections.

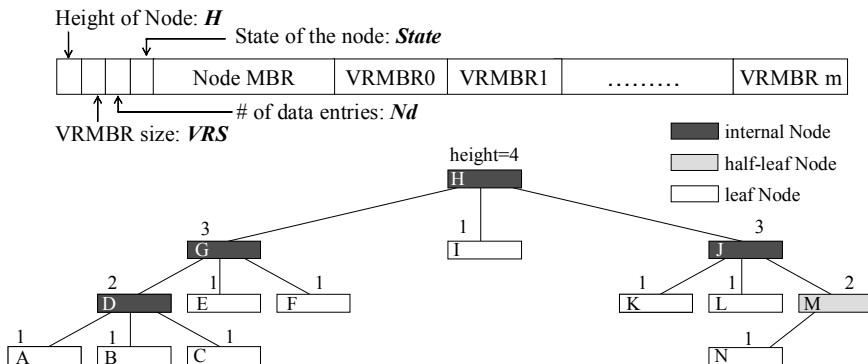


Fig. 3. The MR-tree structure with $diff=2$.

3.2 Node Structure of MR-Tree

The proposed MR-tree is similar to the basic structure of the R-tree. Nonetheless, figure 3 shows that the distinctive features of the MR-tree are that it uses VRMBR instead of MBR and the subtrees of the node of the MR-tree can be unbalanced since the height differences among subtrees can be equal to or less than the unbalance parameter, $diff$. The nodes of the MR-tree are classified into internal nodes, leaf nodes, and half-leaf nodes. The half-leaf nodes have entries for the leaf nodes and for data objects.

Figure 3 also describes the node structure of the MR-tree. The node MBR field encloses all entry MBRs represented by VRMBRs. Coordinates and the size of VRMBR, which is relative to node MBR and of which size can vary according to nodes, are defined as follows.

Definition 1. (Coordinates of VRMBR) Let I and E be n -dimensional MBRs represented by lower and upper bounding coordinates $(l_1, l_2, \dots, l_n, u_1, u_2, \dots, u_n)$. Then, the relative representation of the i -dimensional lower and upper bounding coordinates of E according to I is defined by the relative distance from the i -dimensional lower bounding coordinates of I .

$$VRMBR_l(E.l_i) = \begin{cases} 0 & , \text{if } E.l_i \leq I.l_i \\ I.u_i - I.l_i - 1 & , \text{if } E.l_i \geq I.u_i \\ E.l_i - I.l_i & , \text{otherwise} \end{cases} \quad VRMBR_u(E.u_i) = \begin{cases} 1 & , \text{if } E.u_i \leq I.l_i \\ I.u_i - I.l_i & , \text{if } E.l_i \geq I.u_i \\ E.u_i - I.l_i & , \text{otherwise} \end{cases}$$

Definition 2. (Size of VRMBR) Let I be an n -dimensional node MBR, then number of bits S of a VRMBR in I is defined by the maximum value of $(u_i - l_i + 1)$ among n -dimensional coordinates of I .

$$S_I(VRMBR) = 2n \left(\left\lceil \log(\max_{i=1..n} (u_i - l_i + 1)) \right\rceil \right)$$

Let N be a node of MR-tree, and let S_N and S_{add} be the size of node N and size of additional information fields respectively, the fanout of a node N in MR-tree is then defined by the VRMBR size.

$$Fanout(N) = \left\lfloor \frac{S_N - S_{add}}{S_i(VRMBR)} \right\rfloor$$

In the MR-tree, the size of VRMBR grows when a node is closer to the root. Therefore, when the size of a node MBR and VRMBR in this node are equal, the MR-tree eliminates node MBR and treats VRMBR as MBR.

To increase insertion and deletion performances, the MR-tree maintains three distinctive fields: the height of node (H) field holds the maximum value of height of its child nodes plus one; the number of data entries (Nd) field, used by the half-leaf node, only holds the number of data entries; and the state of node ($State$) field holds the flag value, which is “up” if the node has empty entries, or “undetermined” if the node is full and more than one height difference among its subtrees is equal to $diff$, or “down” if the node is full and all height differences among its subtrees are less than $diff$.

3.3 Searching in MR-Tree

The **search** algorithm of the MR-tree is similar to the R-tree save for two distinct differences. The first is that the MR-tree compares a query MBR with a node MBR when query MBR reaches an index node. Then, when a query MBR overlaps with a node MBR, the MR-tree translates the query MBR to VRMBR by using the node MBR. - And it compares the query VRMBR with VRMBRs in the node to determine whether they overlap. The second is that the MR-tree compares a query MBR with both the VRMBRs of the child nodes and those of the data objects, when it reaches half-leaf nodes.

3.4 Insertion in MR-Tree

The insertion in MR-tree consists of **insert**, **chooseLeaf**, and **AdjustTree** algorithm as with the R-tree [7]. The **insert** algorithm, firstly, invokes the **chooseLeaf** algorithm, which selects a leaf node by using absolute MBRs, and decides whether a leaf node split must propagate upward if it occurs during insertion. Secondly, it transforms a new MBR into a VRMBR and inserts into a chosen leaf, and invokes the **AdjustTree** algorithm. Here, if a chosen leaf is full and the **chooseLeaf** algorithm has decided that the node split need not propagate upward, the chosen leaf splits and becomes the half-leaf node that has an entry for the newly-created leaf node and the data objects. If the **chooseLeaf** algorithm has decided that the node split needs to propagate upward, node splits propagate upward until meeting the internal node in which node splits do not occur.

Algorithm **Insert** and **chooseLeaf**.

```

Insert(in N, in E)
/* N: the root of the MR-tree, E: inserting rectangle */
1. Split_mode := "down";
2. L := ChooseLeaf(N, N, E, Split_mode);
3. if (L has room for another entry) install E in L;
4. elseif (L is a half-leaf node) {
5.     Make new child node CL of L and fill CL
       by all data entries in L;
6.     install E in CL;
7. } else {
8.     LL := splitNode(L,E); //Node L is split into L,LL
9.     if (Split_mode="down") {
10.        Create a new entry ELL;
11.        ELL.child := LL; Add ELL to L;
12.        Adjust ELL.MBR so that it tightly encloses
13.        all entries in LL;
14.    }
15. return AdjustTree;

chooseLeaf(in P, in N, in E, inout Split_mode)
/* P : the parent node of N, N : the current node,
   Split_mode : "up" or "down" or "undetermined" */
1. if (Split_mode<>"up") {
2.     if (N.State="up") Split_mode := "up";
3.     elseif (Split_mode="undetermined" or
              P.State="undetermined")
        {
4.         if (P.H > N.H + 1) Split_mode := "down";
5.         elseif (N is a leaf) Split_mode := "up";
6.         else Split_mode := "undetermined";
        }
    }
7. if (N is a Leaf) return N;
8. else {
9.     for (each entry F in N) {
10.        choose F whose rectangle FI needs least
           enlargement to include E by using
           absolute MBR of F;
        }
11.    if (N is a half-leaf and F is a data entry) {
12.        Split_mode := "down";
13.        return N;
    }
14.    return chooseLeaf(N, F.child, E, Split_mode);
}

```

The **AdjustTree** algorithm, ascending from a leaf node up to the root, adjusts VRMBRs, and recalculates heights, determines the state fields of nodes, propagates node splits upward only if needed. The CR-tree causes some anomalies when adjusting QRMBRs because it uses a lossy compression scheme. The MR-tree, however, does not cause such anomalies because it can always transform VRMBRs into original MBRs.

The **split** algorithm in MR-tree can use the split algorithms used in other R-tree variants including the R-tree and R*-tree [7][8][9]. The only differences are as follows: the heights and VRMBRs of the split node and newly-created one must be recalculated after processing the split; the state fields of both nodes are set to “up” since both nodes have empty entries. In our experiment, we use the linear-cost split algorithm of the R-tree. Figure 4 shows insertion examples.

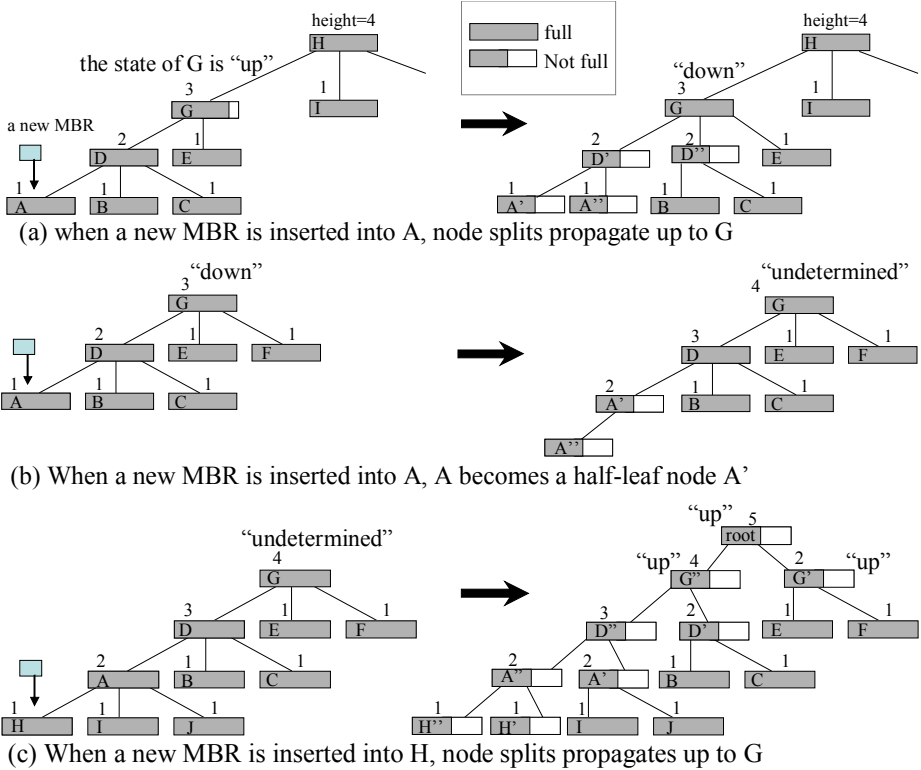


Fig. 4. Insertion examples in MR-tree with $diff=2$.

In figure 4(a), when a new MBR is inserted into node A, the MR-tree splits node A and propagates node splits up to node G, since the state of G is not full. This case is the same with the behavior of insertion in the R-tree. We represent this condition by assigning “up” to the state field of node G. In figure 4(b), node A splits and simply becomes a half-leaf node pointing the newly-created node A’, since the states of G and D are “down”, which means that all height differences among its subtrees is less

than *diff*. We propose this manner of insertion not only to reduce the number of node splits but also to be cache-conscious. A cache miss occurs when the requested data is not in cache. Therefore, assuming that an index node uses a same size of cache line, the number of node accesses must be reduced to be cache-conscious. Compared to the R-tree, this manner of insertion in figure 4(b) is more cache-conscious, since data objects in *B*, *C*, *E* and *F* can be accessed without additional cache miss.

Figure 4(c) shows a worst case in the MR-tree. In this case, when a new MBR is inserted into the node *H*, the MR-tree propagates node splits up to *G*, since the state of *G* is “undetermined” and all height differences among the consecutive nodes on the insertion path are 1. The reasons for using the term “undetermined” is that the MR-tree cannot determine whether node splits propagate upward only through inspecting this node. The MR-tree decides that leaf splits need not propagate upward when it reaches one internal node that is not full, or when it meets the condition that the height difference between one internal node and its parent is more than 1. In figure 4(c), therefore, if a new MBR is inserted into node *F*, the MR-tree splits *F* into two, and *F* becomes a half-leaf, as in the case of figure 4(b). If the MR-tree does not meet these conditions on the insertion path, it can determine whether node splits propagate upward or not only after reaching a leaf. Figure 4(c) illustrates this case.

3.5 Deletion in MR-Tree

The MR-tree uses the **Delete** and **FindLeaf** algorithms used in the R-tree [7]. Compared with the R-tree, however, the **CondenseTree** algorithm in the MR-tree is to be modified to reduce the number of reinsertions. Figure 5 reveals this example.

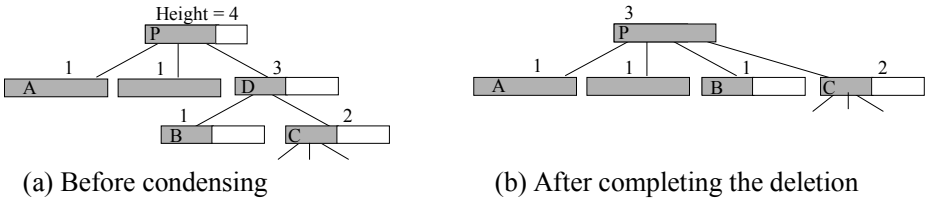


Fig. 5. A deletion example in MR-tree with $diff=2$, when the fan-out is 5.

In the R-tree, when it meets the node in which the number of using entries is less than m (minimum number of entries to be used), the **CondenseTree** algorithm reinserts all its subtrees. Therefore, assuming that figure 5(a) shows the R-tree having 3 for the value of m , all subtrees of node *D* must be reinserted. In the MR-tree, however, the **CondenseTree** algorithm first disconnects *D* from *P*, calculates the height of *P* along with checking node *P* to see how many empty rooms are in *P*, and then examines the height of child nodes of *D*. In figure 5(a), the nodes *B* and *C* can be appended to the node *P* since *P* has two empty rooms and the height differences between *P* and child nodes of *D* are less than the unbalance parameter, *diff*, plus 1. Therefore, without reinsertion, the **CondenseTree** algorithm sets the nodes *B* and *C* to be the VRMBRs in *P* as in figure 5(b) and adjusts the height of *P*. In this way, the MR-tree can reduce entries to be reinserted much further than the R-tree.

4 Experiment

We performed an experimental comparison of the algorithms using GNU's gcc on the Linux operating system. In our case, the platform is a Pentium 4 personal computer (1.6GHz CUP with 256K L2 cache and 128 bytes cache line).

We implemented five two-dimensional main memory index structures: the MR-tree with $diff=1$, $diff=2$, $diff=3$, the ordinary R-tree and the CR-tree. All of these structures use the Lazy Update approach [2]. We use 16 bytes for the size of MBR. For the CR-tree, we use 8-byte rectangles, called QRMBRs, because a too-small QRMBR decreases search performance in our experimental environment. For the purposes of simplicity, the MR-tree uses 4-byte and 8-byte rectangles, called VRMBRs, and 16-byte MBRs.

Owing to the lack of real data, we generated two synthetic data sets, as seen in [1], consisting of one million small rectangles located in the two-dimensional square. One is uniformly distributed in the square whose side length is 1000,000 and the other has the Gaussian distribution around the center point (500,000, 500,000) with a standard deviation of 250, 000. We set the average side size of rectangles to be 1. We randomly arranged data object MBRs in the data sets, since performances of all index structures may be changed according to the insertion sequence. Since most moving objects are in an almost static state most of time [12], we randomly selected only 10,000 data objects after loading one million objects, and changed the positions of these objects.

4.1 Search Performance

In our experiment, we compare the search performance of various index structures in terms of the time spent processing a two-dimensional region query. We generated two sets of 10,000 different query rectangles of the same size per each data set: one is the Gaussian and the other is the Uniform distribution respectively. The sizes of query rectangles per each data set are from 0.01% to 3% of the square.

Figure 6 shows the average elapsed microsecond time spent when the data set of searching various indexes has Gaussian distribution, while figure 7 shows the case when it has Uniform distribution. Observations on these figures include:

- As the node size grows, the search time approaches the minimum quickly and then changes slowly. This trend also applies to increased selectivity. The reason for this trend is that the number of node accesses does not decrease significantly as node size grows, but the number of cache misses increases slightly.
- In comparison to the R-tree and the CR-tree, which form a slower performance group, the MR-trees constitute a faster performance group. The reasons for this result are: the VRMBR scheme increases fanouts of nodes differently according to the VRMBR size, while it does not show any anomaly as discussed in the CR-tree (i.e., entry MBRs, called QRMBRs, are enlarged falsely whenever recalculated); and the manner of insertion in the MR-tree reduces the number of nodes to be accessed, because it eliminates many empty entries in internal nodes.

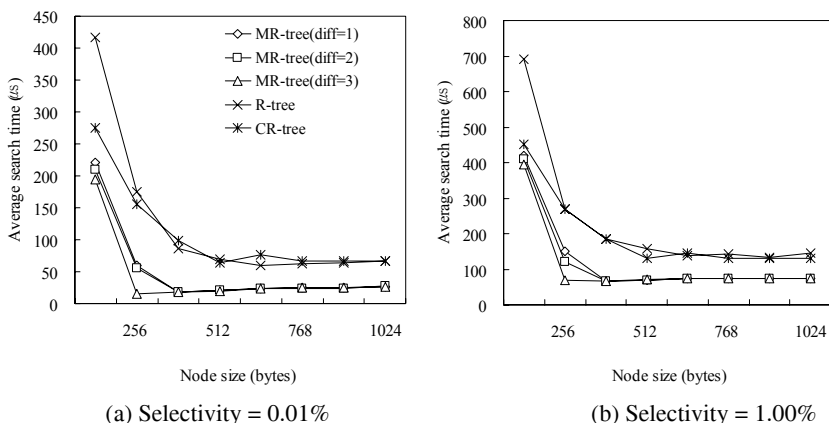


Fig. 6. Search performances (Gaussian distribution with mean = (500,500) and standard deviation = 250).

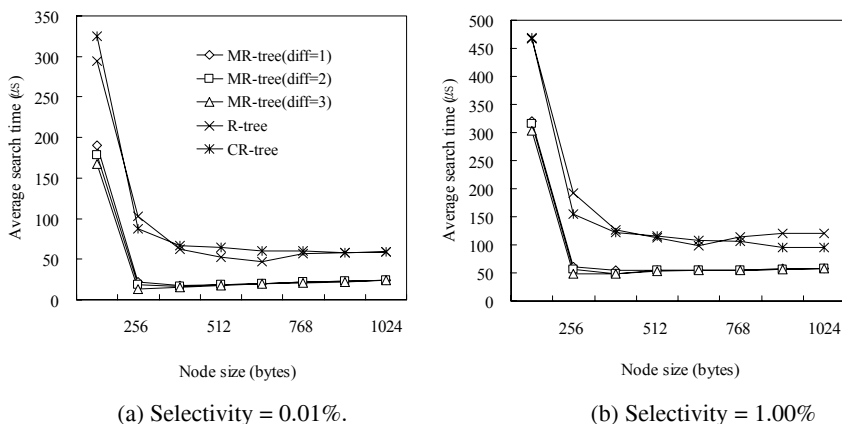


Fig. 7. Search performances (Uniform distribution).

4.2 Update Performance

To measure update performance, we compared the various index structures in terms of the time spent changing the positions of 10,000 data objects. Figure 8(a) and 8(b) show the average elapsed time per update operation. The observations are as follows:

- When comparing update time of indexes, the CR-tree and the R-tree show worse performances than the MR-trees. This is because the MR-trees reduce the number of node splits and reinsertions significantly as shown in figure 9 and figure 10 respectively.
- When comparing the CR-tree and the R-tree, the CR-tree does not show a much better performance, because it spends much time calculating QRMBRs and restoring QRMBRs to original MBRs. In an MR-tree, however, it requires little time to calculate VRMBRs and MBRs from VRMBRs, since the MR-tree represents VRMBR only by relative values.

- When the MR-tree has 3 for the value of the unbalance parameter *diff*, it shows a slightly better update performance in the small node size. The MR-trees, however, show almost same update performances when the node size grows. The reason of this result is that we only updated the 10,000 objects in the experiment and the large node size by itself reduces the number of node splits and reinsertions significantly.

As you can expect from our experimentation, a bigger value for the unbalance parameter, *diff*, will cause better update performance of the MR-tree in an environment where objects update their positions more frequently. On the other hand, search performance can be deteriorated when the value of the unbalance parameter is too big.

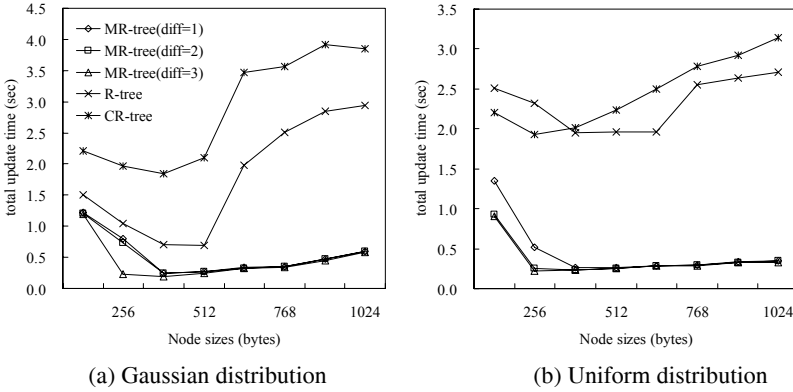


Fig. 8. Update performances.

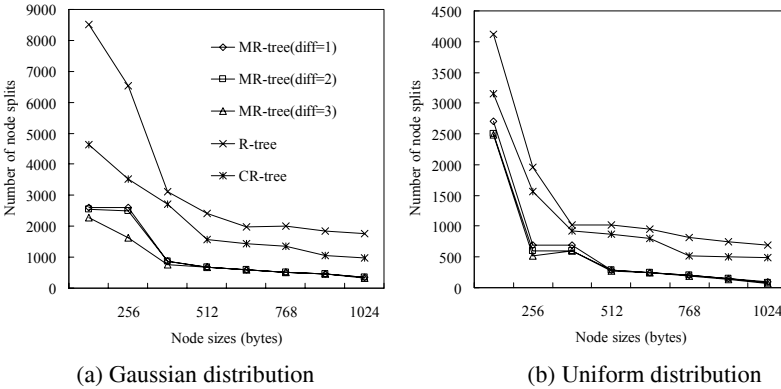


Fig. 9. Number of node splits.

In this paper, we omit the experimentation with multi-dimensional data sets. However, we believe that the MR-tree will also show a good performance in multi-dimension, as like in two-dimension, because the VRMBR scheme is more adaptable and the “unbalance” scheme also reduces the number of node splits and reinsertions

in multi-dimensional environments. We, therefore, conclude that the MR-tree is an efficient n -dimensional index structure for tracking moving objects in mobile GIS applications, because of its reasonable search and update performance.

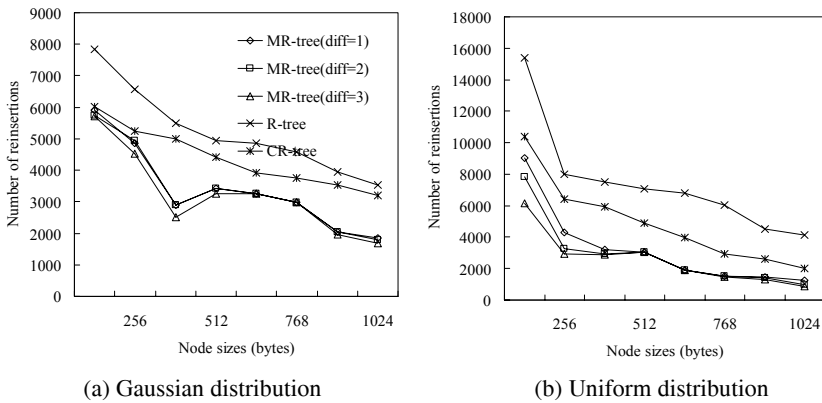


Fig. 10. Number of reinsertions.

5 Conclusion

In this paper, we proposed a modified cache-conscious version of the R-tree, called MR-tree, for main memory data access. The MR-tree uses VRMBRs as index keys, which increase fanouts of nodes in the tree differently according to the VRMBR sizes without any anomalies during insertion and deletion.

To adopt the MR-tree as an index structure for mobile GIS applications, we used the Lazy Update [2] approach to reduce the number of update operations. This approach, however, is of no use when an object moves out of the region of corresponding node. The basic and general solution to solve this problem is to speed up insertion and deletion. In our observation, processing node splits and reinsertions occupy a major part of insertion and deletion time respectively. The MR-tree, therefore, modifies insertion and deletion operations used in the R-tree. We proposed the insertion operation in the MR-tree as follows: the MR-tree propagates node splits upward only if one of the internal nodes on the insertion path has empty rooms, or the state field in one of the internal nodes is “undetermined” and all height differences between the consecutive nodes on the insertion path are 1. Thus, the MR-tree reduces the number of nodes splits significantly. Height differences among the nodes that have the same parent are equal to or less than the unbalance parameter, *diff*. When the MR-tree need not propagate leaf node splits upward, a newly created leaf node simply becomes a child of the split leaf. Then the split node is called the “half-leaf node” because it has entries for data objects in addition to entries for child nodes. We also proposed the deletion operation in the MR-tree as follows: when the number of using entries in a node N is less than the minimum number of entries to be used, the child node of N is appended to the parent of N , when the parent has empty rooms and height differences between the parent and the child node of N is equal to or less than the unbalance parameter, called *diff*, plus one, without reinsertion.

In our experiment, the MR-tree outperformed the R-tree and the CR-tree by the factor of up to two in terms of search time. In update performance, the MR-tree showed better performance in two data sets, particularly in Uniform distribution.

References

1. K. Kim, S. K. and, Cha, K. Kwon, "Optimizing Multidimensional Index Trees for Main Memory Access", *Proceedings of ACM SIGMOD Conference*, 2001, 139-150.
2. D. Kwon, S. J. Lee, and S. Lee. "Indexing the current positions of moving objects using the lazy update R-tree". *3rd International Conference on Mobile Data Management*, Jan 2002.
3. P. Bones, S. Manegold, and M. Kersten, "Database Architecture Optimized for the New Bottleneck: Memory Access", *Proceedings of VLDB Conference*, 1999, 54-65.
4. A. Ailamaki, D. J. DeWitt, M. D. Hill, and D. A. Wood, "DBMSs on a Modern Processor: Where Does Time Go?", *Proceedings of VLDB Conference*, 1999, 267-277.
5. J. Rao, K. A. Ross, "Cache Conscious Indexing for Decision-support in Main Memory", *Proceedings of VLDB Conference*, 1999, 78-89.
6. J. Rao, K. A. Ross, "Making B+-trees Cache Conscious in Main Memory", *Proceedings of ACM SIGMOD Conference*, 2000, 475-486.
7. A. Guttman, "R-tree: A Dynamic Index Structure for Spatial Searching", *Proceedings of ACM SIGMOD Conference*, 1984, 47-57.
8. N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", *Proceedings of ACM SIGMOD Conference*, 1990, 322-331.
9. I. Kamel and C. Faloutsos, "Hilbert R-tree: An Improved R-tree Using Fractals", *Proceedings of VLDB Conference*, 1994, 500-509.
10. Phil Bernstein, et al. "The Asilomar report on database research". SIGMOD Records, 1998, 27(4).
11. O. Wolfson, B. Xu, S. Chamberlaina, and L. Jiang, "Moving objects databases: Issues and solutions", In *Proc. of the 10th Int'l. Conf. on Scientific and Statistical Database Management*, 1998, 111-122.
12. Yuni Xia and, Sunil Prabhakar, "Q+Rtree: Efficient Indexing for Moving Object Databases", In *Proc. Of 8th Int'l Conf. on Database Systems for Advanced Applications (DASFAA)*, 2003
13. V. Gaede and O. Gunher, "Multidimensional access methods", *ACM Computing Surveys*, 1998, 170-231.
14. Ravi Kanth V Kothuri, Siva Ravada, and Daniel Abugov. "Quadtree and R-tree indexes in oracle spatial: a comparison using GIS data", *Proceedings of ACM SIGMOD Conference*, 2002.

Developing Non-proprietary Personalized Maps for Web and Mobile Environments

Julie Doyle¹, Qiang Han¹, Joe Weakliam¹,
Michela Bertolotto¹, and David Wilson²

¹ Department of Computer Science, University College Dublin,
Belfield, Dublin 4, Ireland
{julie.doyle,qiang.han,joe.weakliam,michela.bertolotto}@ucd.ie

² Department of Software and Information Systems,
University of North Carolina at Charlotte, 9201 University City Blvd,
Charlotte, NC 28223, USA
davils@uncc.edu

Abstract. When looking for specific detail within the context of spatial information, users are often faced with the problem of information overload. Moreover, research efforts in responding to such needs, and in general, are impeded by a lack of non-proprietary platforms for development. Personalization is a powerful concept for providing users with precise information that satisfies their current requirements. Personalization techniques are currently in widespread use across the World Wide Web. However, existing Web map servers do not offer much support for personalization in terms of map content. Furthermore, these systems are heavily reliant on proprietary software. We propose a solution to these problems by outlining a prototype system that presents the user with personalized maps containing condensed content. Such a system has been developed using non-proprietary software for both the Web and mobile environments.

1 Introduction

In recent years the Internet has become a major new medium for spatial data. Maps on the Internet help to locate points of interest such as restaurants, hotels or specific addresses, to visualize natural hazards, to plan trips etc. Current GIS applications providing Web maps have obtained considerable success. For instance, the raster map service provider MapQuest.com reported 20,000,000 map requests on a daily basis in the year 2001 [1]. The success can be attributed mainly to the fact that they are easily accessible.

One shortfall of existing GIS lies with the absence of personalization. Personalization is the notion of providing an individual with an informative experience that is tailored to that individual's specific needs at a specific moment in time and is widely employed across the World Wide Web. There is, however, almost a complete lack of content and display personalization in current GIS when rendering maps. When more than one user request maps of the same area, they

are almost always inadvertently presented with the exact same map content and display in terms of fundamental map features. In order to overcome this shortfall systems need to establish either 1) what it is the user may currently be interested in, or 2) what characteristics of maps the user has requested in the past, and hence provide the user with map content that satisfies their current demands.

Current mobile geo-spatial applications do not take full advantage of available spatial information and user context [2]. Mobile users on one hand have to cope with low connection bandwidth, as well as limited computational power and device interfaces, but have the advantages of mobility and real spatial context. In order to provide effective spatial services for the mobile user and to reduce spatial information overload, delivery of information needs to be efficient and on-point. Returning personalized maps to users facilitates these two tasks.

Another major problem with traditional GIS for researchers and developers is the lack of non-proprietary open standard software packages available. This means that large-scale GIS development for these communities can become very difficult, as commercial GIS packages are very expensive. Also problems with interoperability are a major concern when developing a GIS system. Interoperability refers to the capability of autonomous systems to exchange data and to handle processing requests by means of a common understanding of data and requests. GIS users need to be able to share maps over the Web and between different systems. Current GIS systems are also heavily reliant on specific software and hardware, e.g. to view ESRI data, users must purchase ESRI software packages [3].

The motivation for our research stems from resolving issues associated with deploying personalized geo-spatial data across diverse platforms and thus not burdening the user with irrelevant information. The contribution of this paper lies with developing a mechanism for reducing the amount of spatial data that needs to be transferred to the client and hence providing the user with precise detail in relation to map content and display that satisfies their current demands.

The remainder of this paper outlines the various technologies involved in developing our prototype application, including the transmission of an open standard spatial data source and the displaying of this data using open source software in Web-based and mobile-based environments. Section 2 discusses related work in the area of Web personalization. Our system architecture is outlined in section 3. Section 4 examines how map personalization is integrated into our system, while section 5 outlines our system implementation. Section 6 concludes and discusses future work.

2 Related Work

There are a number of approaches for providing tailored content to individuals based on their preferences. These approaches typically incorporate personalization (tailoring of information) that may be accomplished by a collaborative filtering or content-based method (personalization/recommendation technique) which makes use of user profiles (user model) to create its recommendations.

Web Personalization is an extremely powerful concept employed by many e-Commerce applications, e.g. Amazon [4]. There has been significant progress made in the development of personalized systems on the World Wide Web [5]. Map Personalization is a means of delivering specific content, in terms of map features and feature display. The numerous systems, currently available on the World Wide Web, that produce maps, offer little, if any at all, real map personalization. With MapInfoTM's MapXtreme [6], the user can request vector maps and has the capability of altering the map display and performing analyses of various map aspects. However, not only is MapInfoTM's MapXtreme proprietary in terms of making map requests, but it is also geared towards professional users and is not suitable for novice use.

In [7] location-based services (LBS) are used to take into account the spatial, temporal, and contextual characteristics of a user and their interactions and hence provide the user with the most appropriate service based on their local environment. However, such a paper focuses on delivering non-spatial data to the user, e.g. if the user requests information related to restaurants, a list of possible restaurants is returned ordered by opening hours, location, style, etc. Therefore personalization is provided but no map personalization is apparent.

There are several travel and tourism systems that focus on the idea of providing the user with personalized maps by taking other users' interests and preferences into consideration. A user-modeling server that provides services to personalized systems in relation to the analysis of user actions, the representation of assumptions about the user, and the inference of additional assumptions based on domain knowledge and characteristics of similar users is outlined in [8]. These systems provide map personalization to users based on the actions and preferences of other users whereas our prototype proposes ascertaining from an individual's actions and assumptions made about that user what their personal preferences in terms of map content are.

PILGRIM [9] is a mobile system that focuses on the position of the user as a means for returning further information to the user. It makes use of a location broker where a database collects detail about past user locations and links explored in order to establish patterns of spatial usage. PILGRIM is a Recommendation System that is not reliant on the user explicitly rating items. It makes use of the user's current location but is, however, based on the user browsing web pages as opposed to maps. Therefore, personalization is provided in so far that recommendations are made to the user but not regarding map content. Personalization is also evident in other mobile applications such as [10] [11].

One solution for providing map personalization is to record all user map interactions and use the information inherent in these actions as a means of attempting to ascertain the user's interest in terms of map content. In our system, when the user requests a map, only relevant detail is returned in the map to that user, i.e. solely personalized maps are presented to the user. Personalization can be realized by employing customization techniques. Customization allows a user to tick boxes on a map and explicitly state their preferences regarding map con-

tent and display. As a result the features and feature display on the map are altered. Personalization, on the other hand, takes these interactions from the customization phase, and uses these actions to determine the user's preferences implicitly.

3 System Architecture

The following system architecture is proposed for delivering personalized geospatial data using non-proprietary software and to efficiently represent maps in either Web or mobile environments. The scope of the system's functionality ranges from geo-spatial information handling and transmission over slow communication links to personalization techniques and human-computer interaction (HCI). In order to fulfill the requirements of the system in a distributed computing environment, the system architecture (Fig. 1) comprises an n-tier client-server application structure. The architecture encompasses three aspects: 1) a map server, 2) a service provider, and 3) a data deployment component.

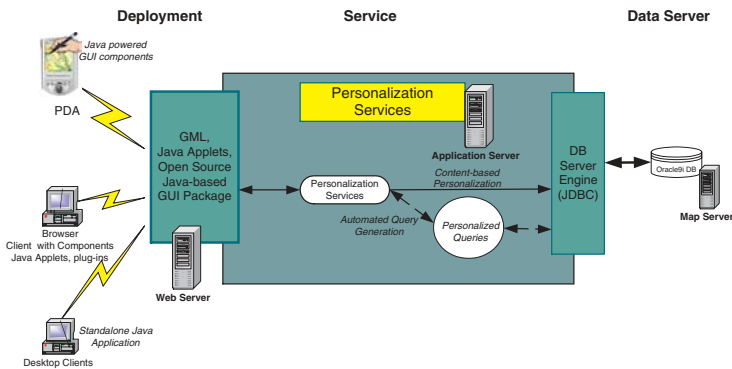


Fig. 1. System Architecture.

3.1 The Data Server

In the data server, geographic maps are represented as vector maps in a map server. Various spatial data components (i.e. geometries, topologies, and semantic content) are represented and managed within one single DBMS (database management system). Currently our system is implemented using Oracle (Spatial) 9i [12] guaranteeing that we conform to OpenGIS Consortium (OGC) specifications from a data-modeling point of view. The data server also houses the user profile. This profile contains detail about the interests of the user regarding map content and is updated after each user session. Using standard SQL (Structured Query Language) the map server provides map contents through an open standard query interface. In a distributed computing environment, we used JDBC to publish our maps in a personalized way through the service provider below.

3.2 The Service Provider

Relying on the data server, the service provider is implemented as a middle-tier application server. It represents an advanced query interface for delivering personalized maps in either Web or mobile environments. Personalized maps display content that is suited to the interests of individual users. All user interactions that have been recorded for each user session are analyzed with the intent of propagating changes to the user's profiles. The personalization service application can effectively capture user preferences within geo-spatial context. Personalization results also need to be integrated with the query mechanisms in geo-spatial databases. In this case, no attempt is made to visualize spatial datasets. Through a deployment layer below personalized maps can be delivered to different hardware platforms in miscellaneous formats.

3.3 Deployment/Presentation Layer

Different clients are connected via the Internet to the data provider's Web server. The Web server then connects through the application server to the data server. In a distributed Web-based and mobile environment clients are classified as PDAs, browser clients and desktop clients. Data in the Web server is generated by personalization applications. These applications are running personalized queries generated through understanding real user requirements during their past interactions with the system. The results of these queries are represented as lists of unformatted geo-spatial contents. These results are then tailored to different requirements based on different users' specifications, e.g. hardware and software platforms etc. These applications include different server scripts, Java applets and GML parsers. In our system we use a non-proprietary GIS GUI package, OpenMapTM [13], to represent map data as a component-based solution. OpenMapTM is a Java-based and open source mapping tool that allows users to develop applications and applets in their own style. This freely available GIS visualization package can be tailored to fit in different specifications by reassembling functional Java Beans.

Based on this flexible system architecture Web-based and mobile users can then query and update the personalized map content through limited connection links.

4 Map Personalization

Personalization is the retrieval of user-specific information for different individuals in varying environments. Map personalization, more specifically, is the generation of map representations with particular content and display. Map personalization involves the following steps:

- Creating a user profile for each user.
- Recording all user interactions with a map generated from the information stored in the user's profile.
- Using these interactions as a means of ascertaining the user's main interests.
- Updating the user's profile on a regular basis to reflect the user's interests.

This form of personalization can be employed in any system that aims to manage the user's main preferences and is applicable to any platform. In the following section, the various aspects involved in providing users with personalized maps are outlined.

4.1 User Profile

In our system, when a user requests a map, rather than returning a fully detailed map containing all possible features, only the most relevant contents with respect to that user are retrieved, i.e. the system is developed to deliver merely essential and appropriate detail to each user. This detail is stored in a user's profile and is a crucial requirement when transmitting maps to low bandwidth devices like PDAs. A distinction needs to be drawn between explicit and implicit user profiling. Explicit profiling [14] is where the user states their preferences by, for example, ticking a series of check boxes indicating features of interest. Implicit profiling [14], on the other hand, is where the system attempts to estimate the user's preferences by monitoring the implicit actions of the user, e.g., if a user zooms in on a feature in a map implies the user may be interested in that feature. Our system makes use of both forms of profiling, but relies on implicit profiling if the user fails to explicitly state their preferences. User profiles are created once the user makes their very first map request and interacts with this map in any manner. The profile stores information concerning those features the user is most interested in and associates a feature weight value with each feature. Therefore the user's profile can be looked upon as a hierarchy of feature-weight pairs and when the user makes a map request only those features with an associated weight value exceeding a minimum threshold value will be returned to the user. The user's profile is constantly evolving in order to preserve an accurate reflection of the user's preferences. Recording all map actions executed by the user during any session can help complete this task.

4.2 Recording User Interactions

Recording user interactions is most important when maintaining an accurate profile of the user's interests. Every single user action executed on the map is recorded in a log file. A list of some possible actions is shown in Table 1. An important distinction needs to be drawn between various actions that can be executed on map features. Map actions fall into one of two categories: *Frame_Actions* and *Feature_Actions*. *Frame_Actions* are executed on all the features visible in the current map frame and include: panning, zooming or re-centering the map frame. *Frame_Actions* do not reveal too much about the user's interests as they usually represent nondescript browsing behavior. *Feature_Actions*, on the other hand, are executed on specific features present in the current map frame and include: toggling a feature on/off, highlighting a feature or performing any spatial query on that feature. *Feature_Actions* are a little more indicative of what map content the user may be interested in or trying to locate.

Table 1. Types of map interactions.

User Action to be executed on the map	Brief Explanation	Action Type
Manual Zoom in	Zooming in by drawing rectangle on the map. The new map generated depends on the area selected within the rectangle and the size of the window.	Frame Action
Panning map in any direction, i.e. N, NE, E, SE, S, SW, W, NW	These eight icons allow the user to pan the map in any direction automatically. When selected the map is panned by a predetermined scale in each direction.	Frame Action
Toggling a layer on/off	The user can turn any layer on or off in the map representation.	Feature Action
X Nearest	User is asked to select a layer type and enter a random number and is then prompted to select a point or rectangle on the map. The specified number of nearest features of that particular type is then highlighted.	Feature Action
Within Xkm	User is asked to select a layer type and a distance value in kilometers. They then select a point or rectangle on the map and all features of that layer type falling within the specified distance are highlighted.	Feature Action

When a user performs actions on the map several things are noted and recorded in the log file for that user session:

1. The id of the map action executed and the id of the map feature(s) involved.
2. The sequence of map actions executed during the entire user session.
3. Those map features present in each consecutive map frame.
4. The time interval in between consecutive map actions.
5. The map features present in the final representation before the user terminates their current session.

The detail that is recorded is then analyzed so that attempts can be made to determine the user's key interests.

4.3 Analyzing User Interactions

Once the user has terminated their current session, the information that has been recorded can then be evaluated offline. Rules are generated for handling events encountered in the log files where certain map actions are executed on certain map features. Rules are also constructed for dealing with certain sequences of actions recorded in the log files. Also if the time interval between any two consecutive user actions exceeds a certain threshold value, the features present in the

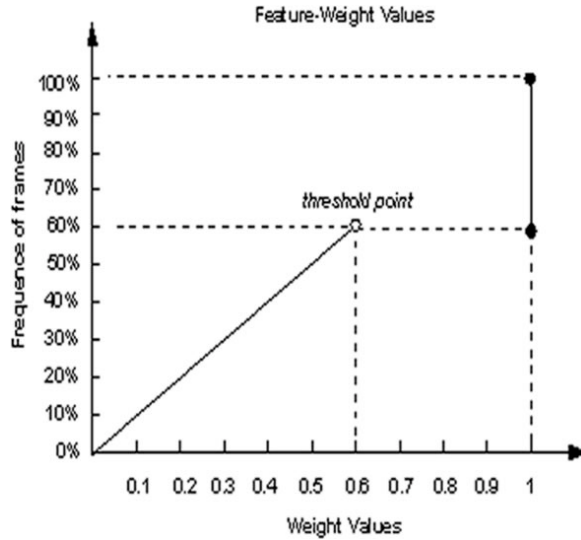


Fig. 2. Graph of feature weight vs. frequency of frames.

map representation before the second action is executed are deemed significant. Counters are set up for every feature present in the initial map returned to the user. For every map frame generated as a result of executing a map action, the counters of all those features present in the resulting frame are incremented by a value of one. If the percentage of frames that a feature appears in exceeds some threshold value then that feature’s weight value is assigned a constant value as this indicates that the user is likely to be interested in this feature (Fig. 2). Once a features weight is assigned a value of one, then that feature will appear in all subsequent maps requested. The above analyses are taken into consideration when updating user profiles.

4.4 Updating User’s Profile

When a user accesses the system for the first time, a default profile is created for that user. All features in this initial default profile are assigned equal weights. Once this first session terminates, the weights of each feature are recalculated, based on interactions performed by the user on map features during this first session. After the second session involving that same user terminates, the new weight values of all the features present in the user’s profile - calculated from the first session - together with the results of analyzing the user’s interactions from the second session, are used to update the weight values in the user’s profile once again. Therefore, in all subsequent sessions where the user requests a map, more personalized maps will be returned to them each time, i.e. the system learns from the user’s behavior in previous sessions to constantly evolve their profile so that relevant representations are always retrieved.

The following flow diagram (Fig. 3) summarizes the major steps involved in providing users with personalized maps.

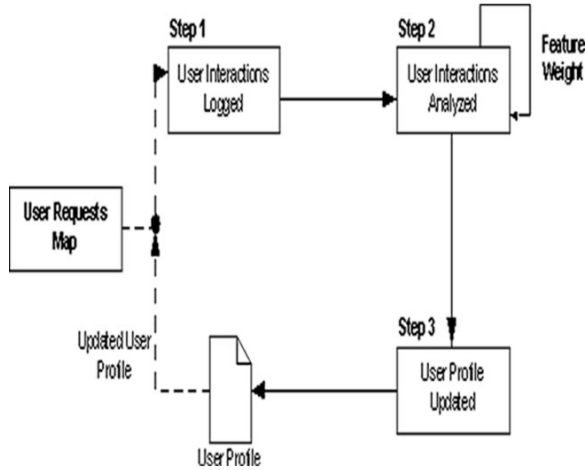


Fig. 3. Flow Diagram.

4.5 Personalization and Mobile Devices

Developing wireless applications can provide user interface designers with a unique set of challenges [15] [16], some of which include limited user interface, limited computational power and limited bandwidth. Incorporating map personalization into our system helps to overcome such problems. Personalized maps contain fewer features than fully detailed maps and hence it is easier to display these maps on the limited screen of a PDA, without the interface appearing too cluttered. PDAs also have limited bandwidth for sending and receiving data. This means that sending large maps to a PDA client may not be feasible, as they would take too long to download. Delivering personalized maps ensures that fully detailed maps are never delivered to the PDA. The problem of limited computational power is overcome in our system as all processing is done on the server side, with the client being used solely to display results.

5 System Implementation

In this section we describe the implementation of our system, outlining the various non-proprietary software components that are used for generating maps with personalized content.

5.1 Personalization Component

Storing the vector maps and user profiles in a remote spatial database (such as Oracle Spatial, Postgres) and using open standard SQL (Structural Query

Language) allows the system to provide personalized spatial querying services. More importantly, spatial and non-spatial data can both be managed and queried under SQL. For each unique user of the system, a unique map representation will be returned to them with respect to the map content and display (Fig. 4). The user is simply prompted to enter their name and the system then retrieves a map containing personalized content. If a user is logging in for the first time they must also compile their profile. Personalization is achieved by extracting all the relevant features from the user's profile, and then selecting the spatial content for rendering each of these features from the database. Only features with a weight value exceeding a minimum threshold value will be returned from the user's profile as those with a weight value that falls below the threshold are deemed insignificant with respect to that user.

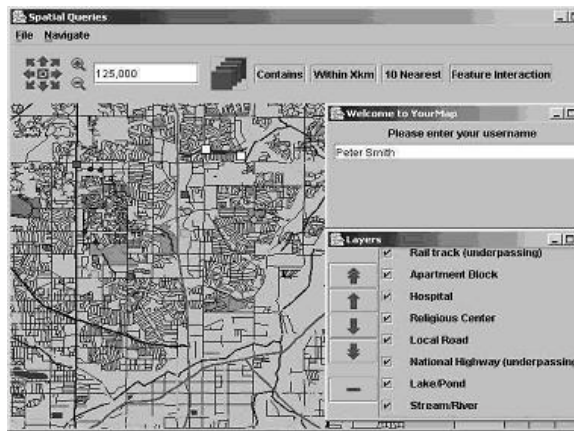


Fig. 4. Screenshot of Personalization Service.

5.2 Web-Based GUI Component

A Web-based component has been implemented using OpenMapTM. Since this GUI is assembled using JavaBeansTM [17], it is fully configurable. This means that users can assemble more tools or simplify the GUI for Web deployment.

Our application allows users to download maps from remote databases and save them in different formats locally (i.e., JPEG, Shape files, TIGER files, GML). Using JDBC interfaces, the system is able to load, query, and manipulate maps represented by a topological structure. Fig. 5 shows a GUI implemented for displaying maps. In this GUI, maps are represented as a series of thematic layers, organized in the navigation panel as a tree-like structure. These layers can be switched on/off based on user requests. A standard toolbar allows users to perform map navigations, (e.g. zoom in/out, pan or scale selection), map edition (e.g., add a new line, delete a polygon, or change the shape of selected entities), and map display (e.g. apply different colour schemas, or line attributes). The addition of a map panel allows users to interact with maps using the mouse.

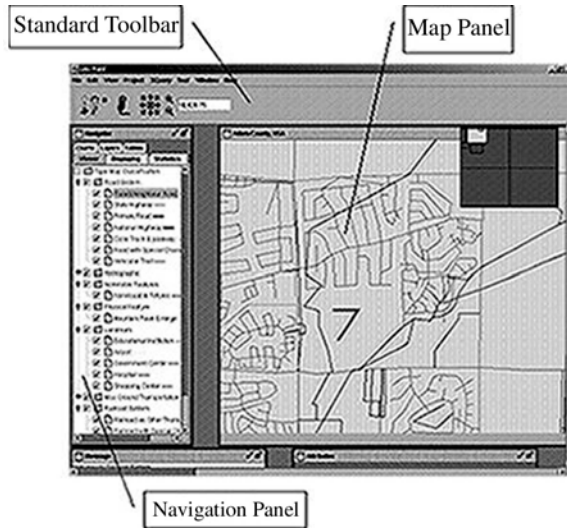


Fig. 5. A Web-based GUI Component.

Our Web-based system permits any user to access (display, query, and download) maps that are stored in a remote spatial database free of charge.

Both the personalization and Web-based GUI components can be deployed to mobile clients. From the remote map server the personalized map contents are transmitted over the wireless network to mobile devices. Using personalization reduces the amount of spatial content that is necessary for displaying a valid map at the client. This is an essential requirement for handheld devices as they are inhibited by a limited bandwidth. OpenMapTM is then used to render the spatial content into a coherent map format.

5.3 Mobile GUI Design

We are currently in the process of deploying the above Web-based GUI component to a PDA. We achieved this by porting OpenMapTM to the PDA which involved redesigning the interface. There are some significant differences between the way you design an interface for the desktop and the way you design it for a PDA. For example, a typical screen size for a PC running Windows is about 800x600. In contrast, the resolution of a PDA is just 240x320, so there is a lot less space to work with. Furthermore, the PDA is in portrait orientation so it is important not to design interfaces that are too wide, as then it will be necessary for the user to use a scrollbar to navigate the page. This is not desirable as it is neither intuitive nor user friendly.

When redesigning the interface for the mobile application it is necessary to take the above points into consideration. The small screen size of the PDA means that not all GUI components from the Web-based system would fit into



Fig. 6. OpenMapTM Interface on a PDA.

a GUI for the mobile-based system, without the interface appearing cluttered. Therefore some GUI components, such as buttons, were removed from the mobile interface. Drop down menus are used instead, as these take up less space. Also, it is necessary to rearrange remaining components on the interface to fit the portrait orientation of the PDA (Fig. 6).

An independent visualisation tool relying on a gml2java program has also been implemented. We are currently exploring the development of this tool on the PDA. This program allows individual users to log onto the system on the client side via their username. The username is sent to the server, which, based on the users profile in the database, chooses the most significant features from the profile and selects these features from the spatial database. This spatial information is then converted into GML (Geography Markup Language) file format [18]. GML is a standard, non-proprietary format that allows geographic information to be exchanged freely between users of different systems - both over the Internet and on mobile platforms. GML is concerned only with describing map content; therefore a method is needed to render maps graphically. GML entities can be rendered as Java objects to display map content. The GML file

is returned to the client device where it is rendered as Java objects within the OpenMap™ interface. The motivation for using GML is that it is an OGC standard and so is open-source and non-proprietary. Moreover, it is based on XML which allows for verification of data integrity, easy integration with non-spatial data and easy transformation.

In terms of content, user requirements and the limitations of PDAs such as limited bandwidth and computational power, it is impossible to deploy full maps to a PDA. However, deploying personalized maps to a PDA is feasible as only specific content is returned to users, hence reducing map size.

6 Conclusion and Future Work

To address the need for map personalization in existing map producing systems, and the dependency of these systems on proprietary software, we have developed a non-proprietary system that provides the user with personalized maps. The system provides personalization through creating and maintaining profiles for each user of the system and analyzing user map interactions. The system architecture is described and the manner in which personalization is delivered to the user is outlined. The system consists of non-proprietary software components and can be deployed to both Web and mobile environments.

We are currently integrating a Location-Based Service into the system to return maps based on the user's actual location using GPS. This means that the user is not burdened with the hassle of stating what area they want to see and a map is returned to them centered on their current location. Future developments include adding more detailed analysis of user mouse movements, e.g. examining the user's positioning of the mouse throughout the whole session, i.e. where the user positions the mouse pointer over a specific feature but does not explicitly query or perform an action on that feature. Further examination of individual user map actions will also be carried out, e.g. when the user performs a manual zoom action (see Table 1) record what features lie at or near the center of the zoom window. We also plan to include dynamic profile updating.

At present the system's performance is restricted when rendering maps containing certain features, namely local roads. The reason for this is due to the fact that this particular feature is by far the largest feature in the database. Local roads constitute about 75 per cent of the total spatial data that is stored at the server. This has serious implications on the overall performance of the system especially with mobile devices. Omitting this one feature from a map reduces significantly the amount of data that needs to be retrieved and then rendered in a map format. Hence, requesting maps without local roads results in maps being returned at a considerably faster rate than those demanding local roads.

There are two approaches for dealing with this issue. The first involves retrieving only a subset of the local roads feature where only those local roads that fall within a certain threshold distance of the user's current location are returned. This decreases the size of the data set considerably as much of the local roads can be disregarded immediately. However, this may result in problems

where the user zooms out of or pans the map to any extent. Progressive Vector Transmission (PVT) is one useful way of overcoming this shortfall where data is transmitted over the network progressively. PVT entails sending the data set in stages in increased levels of detail until the entire data set has been sent. Therefore, when the user requests a map including local roads, a map is rendered at the client in the same amount of time as a map without local roads. However, as the user interacts with the map increased levels of detail appear at the interface as the system sends further data portraying local roads in progressive increments.

Acknowledgments

The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

References

1. M.P. Peterson. Trends in Internet Map Use. In *Proceedings of the 20th ICA/ACI conference, ICMAS 95*, pages 2306–2312, Beijing, 2001.
2. K. Cheverst, N. Davies, A. Friday, and K. Mitchell. Experiences of Developing and Deploying a Context-Aware Tourist Guide: The Lancaster GUIDE Project. In *Proceedings of Mobilcom '00*, Boston, USA, 2000.
3. Esri: <http://www.esri.com/>.
4. Amazon: <http://www.amazon.com/>.
5. N. Hung. Targeting Personalization in ATG. In *ATG White Paper*, 2002.
6. MapInfo: <http://www.mapinfo.com/>.
7. S. Yu, S. Spaccapietra, N. Cullot, and M-A. Aufare. User Profiles in Location-Based Services: Make Humans More Nomadic and Personalized. In *IASTED International Conference on Databases and Applications*, Innsbruck, Austria, 2004.
8. J. Fink and A.Kobsa. User Modeling for Personalized City Tours. *Artificial Intelligence Review*, 18:33–74, 2002.
9. M. Brunato and R. Battiti. PILGRIM: A Location Broker and Mobility-Aware Recommendation System. In *1st IEEE Conference on Pervasive Computing and Communications (PerCom '03)*, Fort Worth, Texas, 2003.
10. C. Anderson, P. Domingos, and D. Weld. Personalizing Web Sites for Mobile Users. In *WWW10 2001*, Hong Kong, 2001. ACM.
11. Weissenberg N, A. Voisard, and R. Gartmann. Using Ontologies in Personalized Mobile Applications. In *ACM GIS*, Washington D.C, USA, 2004.
12. Oracle: <http://www.oracle.com>.
13. Openmap: <http://openmap.bbn.com>.
14. D. Poo, B. Chng, and J-M. Goh. A Hybrid Approach for User Profiling. In *36th Hawaii International Conference on System Sciences (HICSS '03)*, Hawaii, 2003.
15. C. Borntrager, K. Cheverst, N. Davies, A. Dix, A. Friday, and J. Seitz. Experiments with Multi-Modal Interfaces in a Context-Aware City Guide. In *Mobile HCI '03*, Udine, Italy, 2003.
16. F. Bellotti, R. Berta, A. De Gloria, and M. Margarone. Supporting Efficient Design of Mobile HCI. In *Mobile HCI '03*, Udine, Italy, 2003.
17. Javabeans: <http://www.java.sun.com/products/javabeans>.
18. Opengis Consortium Inc.: <http://www.opengis.org>.

Labeling Dense Maps for Location-Based Services

Qing-Nian Zhang^{1,2}

¹ School of Geography, Zhongshan University, 510275, Guangzhou, China

² GIS Centre, Lund University, SE-223 62 LUND, Sweden
zhangstudio@sohu.com

Abstract. Spatial information is often presented as maps in location-based services, which makes it necessary to label cartographic features in real time. Features may be dense all over the map, or in certain areas. Owing to the limited free spaces, it is always difficult to label dense features. Aiming to utilize free spaces efficiently, this paper proposed a density-based method of labelling dense features. The method placed labels of dense features earlier than sparse ones, so that free spaces were allotted to dense features before consumed by sparse features. An efficient algorithm was developed for map labelling in real time. We implemented this method in a Java environment. A case study shows sound cartographic results and acceptable efficiency of the labelling.

1 Introduction

Location-based services have been available in recent years, with the continuing advances in areas such as wireless communications and positioning technologies. Wireless carriers can use their mobile devices to communicate with service providers to access remote databases. In location-based services, a number of spatial information can be queried and offered. For instance, the nearest business or service, the dynamic distribution of tracked resource, etc.

Spatial information is often presented as *real-time maps* in location-based services [22]. Since location-based services involve in tracking resources with dynamic distribution, maps have to be created, updated and delivered in real time. That means maps cannot be created and stored in databases in advance and then delivered to users later. Since it is usually complicated to process spatial information, real-time map creation and updating is not a trivial work. What's more, map requests may lead to additional processing so as to create a usable map. For instance, users may want to combine data from several sources, which may involve in real-time generalization and data integration [7, 9, 14].

Labels are text description of map content, which should be added in real time in location-based services. The labels should be easy to read and follow basic cartographic rules. These rules have to be analytically defined and implemented in an efficient computer program so as to label maps in real time. A number of labelling algorithms have been developed to automate map design and production. Some of them are efficient and can be used in real-time environment [14, 15].

Features may be dense all over the map, or in certain areas. For instance, there are always clusters of cities in regional maps. Owing to the limited free spaces, it is always difficult to label dense features. This study aims to develop and test a method for dense map labelling in real time.

This paper starts with a short review of previous work on automated map labelling. A method for dense map labelling is thereafter described. It generates candidate positions in a continuous space, and place labels in densest area first. This algorithm was implemented in a Java environment and evaluated in a case study, as described in the following sections.

2 Previous Work on Label Placement

Label placement has been an important issue in cartography, GIS, computational geometry, computer aided design (CAD), and others areas in the past decades. A large amount of efforts have been devoted to automatic label placement from researchers with different background. As is already known, labels have great linguistic, practical, technical, and aesthetic importance, and good labelling should conform to a number of cartographic and aesthetic rules [10]. However, only several rules can be easily formalized and implemented in a computer program. Of them, two rules are regarded as very important ones and are widely implemented in labelling algorithms. That is, least disturbance of map content and unambiguous label-feature association [18, 15].

Human cartographers place a label inside a continuous area around a feature. That is, a rectangular area around a point feature, a narrow strip along a line feature, and one or several strips inside an area feature (the text is often stretched to cover the main body of the area) [10, 12]. However, the continuous area of a label is substituted by a fixed number of candidate positions in most labelling algorithms, and the label can only be placed at one of these positions [1, 5]. For instance, the label of a point feature is usually placed at one of eight positions around the point. Such an approach is named as fixed-position model in point labelling. Although this approach is easy to implement, it is not always effective to find a solution. A more effective approach is to include all candidate positions of a label. That is, to find a solution in a continuous area. Several algorithms to label points in continuous solution space have been developed in recent years [15, 14]. For point labelling, an approach with continuous solution space is also named as slider model.

Map labelling is a NP-hard problem, and a number of optimization techniques have been investigated to search for an optimal solution [1, 4]. As two classes of optimisation techniques, gradient descent and simulated annealing were implemented in early studies and compared with existing algorithms [1]. A hybrid force-based simulated annealing algorithm was also developed for label number maximization problem [4]. In the hybrid algorithm, force-based model aimed at a minimum energy configuration, and simulated annealing was employed to escape from local minima. In recent years, genetic algorithm was introduced into the area of map labeling [16, 2]. Test showed that GA algorithm performed better than other algorithms, including simulated annealing [2].

Focusing on maximizing the number or size of labels, several approximation algorithms have been proposed to find a practical solution of map labelling. Of them, a position-packing algorithm was developed to compute the solution by packing the candidate positions of labels [6]. Firstly, Two positions among all four possible positions of a point label were removed according to conflict detection. Then conflict partners were further removed to find a labelling solution. The algorithm computed a

valid labelling of at least half the optimal size in $O(n \log n)$ time. A hybrid algorithm was also developed to combine the approximation algorithm with heuristics to guarantee the quality of optimal approximation [18]. Recently, an algorithm of point labelling in slider model was proposed, which labelled at least half of the point features in $O(n \log n)$ time [15]. The algorithm was further extended to take label-feature overlap into account for point labelling in slider model [14]. It has been proved that no polynomial time approximation algorithm with a better quality guarantee exists if $P \neq NP$ [6].

3 Requirements of Dense-Feature Name Placement

Generally speaking, most maps contains only limited number of features in location-based services. However, they may also contain dense features in some cases. For instance, a user may zoom out to get an overview map, or create a comprehensive map containing all related information. In such cases, cartographic features may be dense all over the map or in certain area.

It is often difficult to label dense maps owing to the limited free spaces. On dense-feature maps, the intervals between features are quite small, which means ideal placement is not always possible. In such cases, a great effort is devoted to find an acceptable placement rather than an ideal one. Sometimes, labels are even omitted when their inclusion in the solution is impossible.

The difficulty of dense-maps labeling depends, to a large extent, on how to effectively utilize the limited free spaces. Number of candidate positions, rules and orders for label placement are three important factors that influence the utilization of free spaces. Firstly, Labels can be placed in either fixed-position model or slider model. Since slider model contains all possible positions besides these contained in fixed-position model, the probability to find a labeling solution in slider model is larger than that in fixed-position model. Thus, it is reasonable to employ the slider model for dense map labeling.

Secondly, labels are required to satisfy a number of cartographic rules. Many cartographic rules may be considered in map labelling. Some rules are prerequisite of acceptable labelling, while others influence only minor aspects of labelling. Generally speaking, the more rules are considered in map labelling, the less free space can be utilized. For instance, if we don not allow label-feature overlap, available space free of label-label conflict has to be further reduced. For this reason, we consider only two basic rules in this study. That is, label-label, and label-feature conflict must be avoided; the association between labels and features should be unambiguous.

Lastly, labels may be placed sequentially or in specialized orders. For instance, the label of the left-most point feature may be selected first, and placed at the left-most candidate position [15]. In this way, labels are packed to the left side of the point feature so that as many labels as possible can be placed. Since top-right position is preferred for point labelling, it will be better to place labels in right-most order. Labels may also be placed randomly. That is, randomly select and place each label until all labels are placed. However, none of these labeling orders is good at maximizing the number of successfully placed labels. In these ways, labels with large free spaces may be placed first and free spaces near dense features may be consumed, which

make it difficult to label these dense features. Thus, labels of dense features should be placed before their free spaces are consumed. In order to place as many labels as possible in dense area, we place the dense-most label first in this study.

4 Densest First Map Labelling

Our method labels points and lines in slider model, where labels of dense features are placed first. The method consists of three stages. That is, generating candidate positions, ordering labels by density, and placing labels ordered by density.

4.1 Generating and Reducing Solution Spaces

This stage computes a solution space for each feature to be labelled according to the detection of label-feature conflict. This study adopts the method described in [22], which labels both point and line features in continuous solution spaces. Label size may be different among different feature classes and features of different importance. This sub-section summarizes the method of generating and reducing solution spaces for point and line labelling. For a detailed description, see [22].

For a point feature, the initial solution space of a label is a rectangle around the point (e.g., the upper rectangle in Figure 1a); for a line feature, a set of long segments of the line whose length are larger than that of the label. For a point, the label can be placed anywhere on the rectangle (e.g., the lower rectangle in Figure 1a); for a line, anywhere on the segments. However, at some of the positions, labels may conflict with map content, which means the initial solution space has to be reduced. Take Figure 1 as an example, if the label is placed at the bottom-left corner of the rectangle (the initial solution space), it will conflict with a road, and thus the solution space is reduced to a part of the rectangle (Figure 1b).

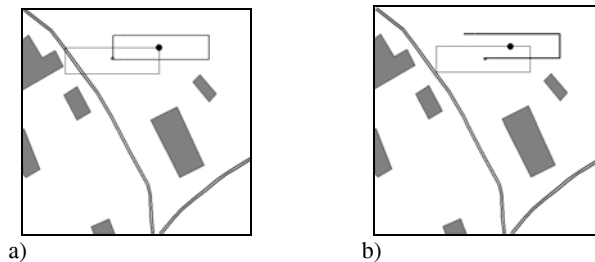


Fig. 1. The initial (left) and reduced solution space (right) of a point label.

Possible label-feature conflicts are detected and avoided using “range box”. For a point feature, four range boxes are created around the four edges of the rectangle around the point feature. Among them, two horizontal range boxes are twice the width of the text label, and the same height as the label; two vertical range boxes are the same width as the label, and twice the height as the label (Figure 2). For a line feature, one oriented range box is created along each of the selected long segments of the line. Each oriented range box is as long as the corresponding segment.

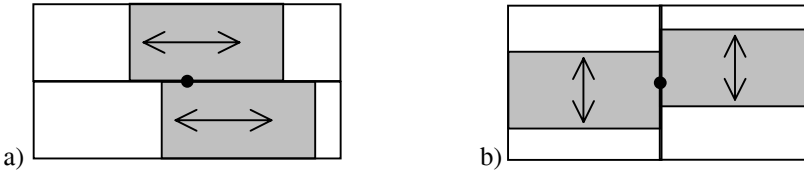


Fig. 2. Two horizontal range boxes and two vertical ones represent the area a point label may move inside.

Range boxes are reduced if they overlap one or more cartographic features. If overlap exists, a range box may be broken into two or more parts. Among them, small parts less than label size and all parts in conflict are removed. For point labelling, a range box is at most twice the size of the label, and thus at most one part is remained. For line labelling, a range box may be twice more large as the label size, and thus decomposed into one or more new range boxes.

Range boxes generation and reduction for point and line features are processed in a similar way, except the latter is dealt with in a rotated coordinate system. Furthermore, lines are simplified to be a set of line segments using the Douglas-Peucker algorithm [3] before computing their range boxes for label placement. In this way, we place a straight text label along a section with gentle bend, taking it as a straight segment.

The final solution space of a label can be easily derived from the reduced range boxes. Firstly, compute the centerline of the range box. Then shrink both ends of the centerline inward half of the size of the label. The reduced centerline contains all the positions where the label can be placed, in no conflict with any cartographic features.

4.2 Ordering Features by Density

We place labels for dense features first, and sparse features later. In this way, the labels for sparse features will not consume the limited free spaces that may be used to label dense features, since dense features are labeled earlier than sparse features. However, the planar partition by feature density is somehow time-consuming. Thus, a density indicator is used instead of density parameter itself.

Obviously, in sparse area, distances between features are relatively big. The initial range boxes attached to sparse features overlap relative fewer map contents. In contrast, initial range boxes of dense features often overlap heavily with each other and other features. In the above stage, range boxes are reduced to avoid label-feature overlap. Obviously, the more a range box is reduced, the more crowded map area it belongs to. Thus, the reduction ratio of range boxes is an effective indicator of label-feature overlap degree, and accordingly, cartographic feature density. Since the candidate labeling area of each feature to be labeled is represented by several range boxes, the density difference is indicated by the reduction ratio of the sum of the areas of the range boxes. For example, the sum of the areas of the initial range boxes of a point label is eight times of the label size. If the sum of the areas of the reduced range boxes of the label is two times of the label size, the reduction ratio will be 0.25.

Similar to the reduction ratio of the area of range boxes, the reduction ratio of the length of the solution space also indicates density differences. As mentioned in Section 4.1, the final solution space is computed as a reduced centerline of the range box.

The reduction ratio of the length of the solution space indicates more accurately the density difference than the reduction ratio of the area of range boxes do. However, it cannot tell the difference between the case when solution space is reduced to null and that to a point. Therefore, in our study, we employed the area reduction ratio to indicate density difference.

4.3 Labeling the Densest Feature

The method to select labeling position is straightforward in this study. It consists of two steps, i.e., select a reduced range box, and a position on the centerline of the range box. For point labeling, since cartographers prefer to place the label to the top-right of a point feature, we select the reduced top range box first. If it is not available (the top range box is removed), the reduced right range box is then selected (thirdly the bottom one and lastly the left one). For line labeling, we select the reduced longest range box.

Once a reduced range box is selected, a position on the centerline of the range box is then selected to place the label. For point labeling, we select the right (for horizontal centerline) or top end (for vertical centerline) to place the label. For line labeling, we place the label at the midpoint of the reduced centerline.

4.4 Updating Solution Spaces

After a label is placed, some of the free space is consumed. Therefore, the range boxes have to be reduced when they conflict with the newly placed text label. The processing is similar to section 4.1, with only the newly placed label as obstacles that are not allowed being overlapped.

In our study, the density order is not adjusted according to the updated range boxes and solution spaces. That means we sort the labels only once before the first label is placed. After updating the solution space, the second feature in the ordered list will be picked out as the second densest feature and accordingly labeled. This process is executed iteratively for all features to be labeled.

4.5 Considering Important Features

The method can be further refined to consider important features. Cartographic features have different importance, and important features (e.g., landmark features) should be labeled as far as possible. A straightforward method is to place important labels first in case free space being consumed by other features. However, free space cannot be utilized effectively in this way, and unsuitable for dense map labeling. On the other hand, some features are not important, and their labels can be omitted if free space is unavailable. These features can be labeled in a separate run after other features have been labeled.

In this study, dense feature will be labeled first in most cases. However, if features' importance is quite different, they will be divided into several groups, i.e., important feature, ordinary feature, and secondary features. For important features, if its reduction ratio is larger than 50 percent, it will be labeled first. That means enlarging the

ratio to almost 100 percent so as to process it first. In this way, the probability that important features fail to be labeled decreases. For secondary features, they are labeled in a separate run after important and ordinary features have been labeled.

4.6 Computational Complexity

The algorithm consists of three time-consuming parts. That is, search for possible conflicting features for a range box, reduce the range box, and order range boxes by density. The first part costs $O(n \log n)$ time, the second $O(nm)$ time, and the third $O(n \log n)$ time. Therefore, the total computational complexity of the algorithm is $O(n \log n + nm)$ time, where n is the number of point features to be labeled, and m is the number of features that should not be overlapped (obstacles). On average the algorithm is quite fast. There are only a few features among the obstacles conflicting with the range box of the label, thus m is (on average) small.

5 Experimental Results

5.1 Implementation Environment

The map labelling method was implemented as part of an application for generalization and integration of cartographic data [11, 7 9]. It was designed as a Java API based on the open source Java packages, JTS Topology Suite (JTS) and JTS Unified Mapping Platform (JUMP). Both packages are from Vivid Solutions [17]. JTS conforms to the *Simple Features Specification for SQL* (by Open GIS consortium) and it contains robust implementations of the most fundamental spatial algorithms (in 2D). JUMP contains import and export functionality for GML data, as well as a viewer.

The current implementation is manually triggered from JUMP and then the following is performed:

- (i) Perform a request for map data from a WFS server (Web Feature Service) [19].
- (ii) Parse the GML-data (Geographic Markup Language) [8] from the WFS server into object instances in the Java environment. Several feature classes can be obtained in one request.
- (iii) Linear features that are representing one physical entity but are divided into several features in the cartographic data are merged. The reason is that each linear feature should only have one label.
- (iii) The map labelling is performed according to the methodology section above. Furthermore, labels are constrained to be inside the map boundary. In this part several functions from JTS are used.
- (iv) The map is displayed in JUMP.

5.2 A Case Study

We carried out a case study querying map data from a WFS server at the Finnish Geodetic Institute. We requested feature classes as named locations (text data), roads, railways, buildings, lakes, and rivers. However, for legibility reason, only roads and named locations are employed in the case study. We wanted to label the names for the

named locations (point labelling) and the roads (line labelling) using name attributes. No predefined positions for road name placement were available. The named locations, on the other hand, had predefined positions. However, in this case study we treated the predefined position as a point object and placed a label around it. Labels for point and line features are placed in a single process. Labels of named locations were not allowed to overlap other named locations; labels of roads were not allowed to obscure named locations and roads; labels are not allowed to overlap each other.

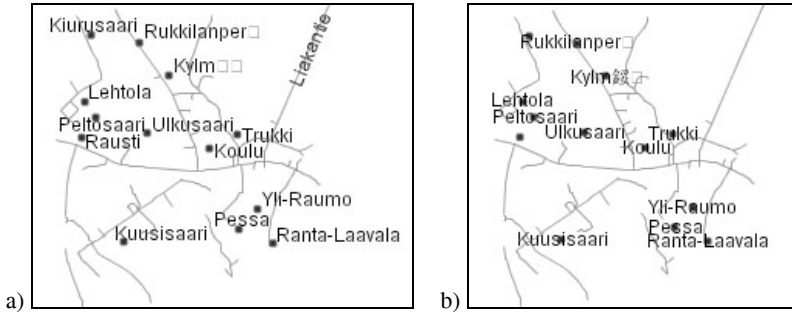


Fig. 3. (a) Labels positioned by our method. (b) Labels added by the built in functionality in JUMP. © National Land Survey of Finland.

Our results are presented in Figure 3a. All the 13 named locations were labelled, and one road was labelled. We found that most of the labels were placed at suitable positions without overlapping specified feature classes. Since our routine merged named roads from segments, at most one label was attached to each road. However, only one line segment was labelled, since segments should be long enough to attach straight text description of lines in our method.

In comparison to the map created by the labelling function provided in JUMP (Figure 3b), the label quality is, to a large extent, improved. The default function labelled 11 of 13 named locations. We found the named locations were not properly labelled, and two of them were not labelled owing to the inefficiently utilizing of free spaces. For legibility reason, roads were not labelled in Figure 3b, since the built-in function labels each segment of a line, and made the map hard to read. Since the method consider only label-label conflict within the same feature class, label-label conflicts between different feature classes and label-feature conflicts widely exist. The repeated labelling of different segments of the same road made the labels even more crowded.

In European style maps, road names are labelled inside the road symbol. However, roads were presented as single lines in this case study. Otherwise, displacement of features has to be dealt with. When roads are symbolized as cased lines, road symbol will overlap nearby buildings and other features. Therefore, displacement is necessary to remove the conflict between cartographic features. However, this study focused on label placement, and left displacement to other routines.

The CPU time for the labelling of the map in Figure 3a was 78 milliseconds on a personal computer of 1.8 GHz, Pentium IV processor and 256M RAM. This is fairly acceptable in our study. In a real mobile application, the server machine is more powerful than our personal computer.

5.3 Other Case Study

We also carried out other case studies. These studies were based on free cartographic data downloaded from the University of Iowa. One of the experimental results was presented in Figure 4. In this figure, cities at the map centre were relatively dense. We want to label the cities and roads, and labels of roads are not allowed to overlap cities. Since roads' names are not available, we used the county names attached to the roads to label them. All points and long segments were labelled at proper positions. The labelling process takes 500 milliseconds.

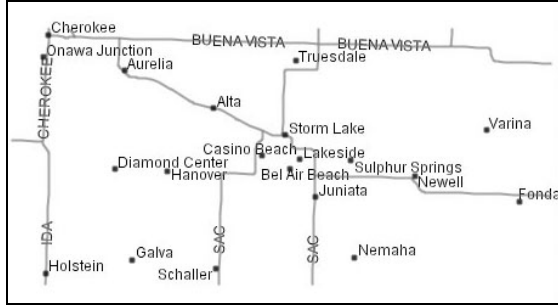


Fig. 4. An experimental labelling result about Iowa. Source: ftp://ftp.igsb.uiowa.edu/gis_library/IA_state/Infrastructure.

6 Discussions

Straight text for line labelling is acceptable and effective when the lines are fairly straight. Since text labels are placed inside or along line symbols, long sections have to be found along the line. However, this condition may fail to be satisfied, and a labelling position along the line may fail to be found. If a number of cartographic features are not allowed to overlap line labels, the labelling space will be further constrained. When long segments cross cartographic features that can not be overlapped, they may be decomposed into several segments too short to contain line labels. In such cases, curved text may be necessary, or overlap constraints have to be relaxed.

When user panning on the map, the method has to compute a new configuration of label placement, which makes the panning not smooth. However, since cartographic features can move smoothly when user pan on the map, the re-arrangement of label configuration will only slightly discretize the panning.

Ideally, the map labelling should be implemented in a system architecture for real-time map services. In our current implementation, map labelling is triggered manually inside JUMP environment. This implementation remains to be transplanted to a real-time map service architecture.

7 Conclusions

In this paper we presented a method of dense-map labelling. The method was based on labelling dense feature earlier than sparse ones in a continuous solution space, with

a constraint to disturb map content as little as possible. A case study revealed that the method gave sound cartographic results and acceptable efficiency.

Acknowledgements

This project was financed by the International Office and the GIS Centre at Lund University. The study was also partly supported by National Science Foundation of China under grant No. 40101024, and partly supported by the '985 Project' of GIS and Remote Sensing for Geosciences from the Chinese Education Department. I would like to thank Lars Harrie for constructive comments, and GiMoDig-project (especially Tommi Koivula for providing routines for parsing the GML-data). Data for the case study was kindly provided by the National Land Survey of Finland.

References

1. Christensen, J., Marks, J., Shieber, S.: An Empirical Study of Algorithms for Point-Feature Label Placement. *ACM Transactions on Graphics*. 14 (1995) 203-232
2. Dijk, V. S., Thierens, D., Berg, D. M.: Using Genetic Algorithms for Solving Hard Problems in GIS. *Geoinformatica*. 6 (2002) 381-413.
3. Douglas, D. H., Peucker, T. K.: Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *The Canadian Cartographer*. 10 (1973) 112-122
4. Ebner, D., Klau, W. G., Weiskircher, R.: Force-Based Label Number Maximization. <http://www.ads.tuwien.ac.at> (2003)
5. Edmondson, S., Christensen, J., Marks, J., Shieber, S.: A General Cartographic Labeling Algorithm. *Cartographica*. 33 (1997) 13-23
6. Formann, M., Wagner, E.: A Packing Problem with Applications to Lettering of Maps. In: *Proc. 7th Ann. ACM Sympos. Comput. Geom.* (1991) 281-288
7. GiMoDig, Geospatial Info-Mobility Service by Real-Time Data-Integration and Generalization. <http://gimodig.fgi.fi/> (2004)
8. GML, Geographic Markup Language. <http://www.opengis.org/techno/documents/02-023r4.pdf> (2003)
9. Harrie, L., Johansson, M.: Real-Time Data Generalization and Integration Using Java. *Geoforum Perspektiv* (2003) 29-34
10. Imhof, E.: Positioning Names on Maps. *The American Cartographer*. 2 (1975) 128-144
11. Lehto, L.: GiMoDig System Architecture. <http://gimodig.fgi.fi/deliverables.php> (2003)
12. Robinson, A. H., Morrison, J. L., Muehrcke, P. C., Kimerling, A. J., Guptill, S. C.: Elements of Cartography. 6th edn. John Wiley & Sons (1995)
13. Sarjakoski, T., Lehto, L.: Mobile Map Services Based on an Open System Architecture. In: *Proceedings of the 21st International Cartographic Conference*, Durban, South Africa, (2003) 1107-1113
14. Strijk, T., van Kreveld, M.: Practical Extensions of Point Labeling in the Slider Model. *Geoinformatica*. 6 (2002) 181-197
15. van Kreveld, M., Strijk, T., Wolff, A.: Point Labeling with Sliding Labels. *Computational Geometry*. 13 (1999) 21-47
16. Verner, V. O., Wainwright, L. R., Schoenefeld, D. A.: Placing Text Labels on Maps and Diagrams Using Genetic Algorithms with Masking. *Inform Journal on Computing*. 9 (1997) 266-275

17. Vivid Solutions: Java Topology Suite. <http://www.vividsolutions.com/jts/jtshome.htm> (2004)
18. Wagner F., Wolff A.: A Practical Map Labeling Algorithm. *Computational Geometry*. 7 (1997) 387-404
19. WFS, Web Feature Service Implementation Specification. <http://www.opengis.org/techno/specs/02-058.pdf> (2003)
20. Wolff, A., Knipping, L., van Kreveld, M.: A Simple and Efficient Algorithm for High-Quality Line Labeling. In: *Proceedings of 15th European Workshop on Computational Geometry (CG '99)*, Sophia-Antipolis, (1999) 93-96
21. Wolff, A.: The Map-Labeling Bibliography. <http://i11www.ira.uka.de/~awolff/map-labeling/bibliography/> (2004)
22. Zhang Q., Harrie L.: Real-Time Map Labelling for Mobile Applications. *Computers, Environment and Urban Systems*. Accepted.
23. Zoraster, S.: Practical Results Using Simulated Annealing for Point Feature Label Placement. *Cartography and Geographic Information Systems*. 24 (1997) 228-238

Mobile SeoulSearch: A Web-Based Mobile Regional Information Retrieval System Utilizing Location Information

Yong-Jin Kwon and Do-Hyoung Kim

Department of Information & Telecommunication Engineering
Hankuk Aviation University, Koyang-shi, Kyounggi-do, 412-791, Korea
{yjkwon,dhkim}@tikwon.hangkong.ac.kr

Abstract. We developed a regional information retrieval system for the mobile environment, called “Mobile SeoulSearch”. This system provides regional information related to the specific position where a mobile user is located. Here the regional information is derived from the Web space. For utilizing the features of mobile environment, and coping with limitations of user interface in mobile devices, several methods are proposed: Firstly, using user’s location information which is obtained in real-time, a method that regional information related to the mobile user’s specific location is provided in real-time to mobile devices, is proposed. Using this method, the mobility of mobile devices is maximized, and the limitation of input methods on the devices is coped with. Secondly, a layered and reconfigurable graphical user interface similar to CardLayout in Java, is proposed so that lots of regional information is displayed efficiently in spite of very poor user interface in mobile devices. Finally, methods for gathering Web pages and indexing regional terms, are proposed to provide mobile users with the regional information effectively. The efficiency of these methods are verified through the development of “Mobile SeoulSearch”.

1 Introduction

The convergence of various technologies, including the Internet technology, wireless communication technology, location technology, and GIS(Geographical Information System) technology have given rise to new types of computing environment. In particular, with the development of wireless Internet and mobile devices such as PDA(Personal Digital Assistant) and HPC(Handheld PC), various fields of application services have been available, and users have been able to gain access to necessary information at any place and at any time. However, applications suitable for the mobile environment are not sufficient in spite of the rapid growth of mobile environment.

One of the most important characteristics in mobile environment is mobility. Generally, mobile devices have the small size of screens and the special input devices, such as a touch screen and a pen mouse, to support mobility. However,

these characteristics of mobile devices also work as a limitation of user interface in mobile devices.

In this paper, we developed a mobile regional information retrieval system, called “Mobile SeoulSearch”, which is a mobile version of “KG21Search” [5,6]. This system provides a layered and reconfigurable graphic user interface similar to CardLayout[14] in Java, so that lots of regional information is displayed efficiently in spite of very poor user interface in mobile devices. In addition, the system provides regional information related to the specific position where a mobile user is located in real-time, so that the mobility of mobile devices is maximized, and the limitations of input methods on the devices are overcome.

The rest of this paper is organized as follows. Section 2 overviews an existing regional information retrieval system, “KG21Search”. Section 3 presents an overview and the configuration of “Mobile SeoulSearch”. Section 4 describes details of methods proposed in this paper to develop “Mobile SeoulSearch”. Finally, Section 5 contains the conclusion and future work.

2 Existing Regional Information Retrieval System

A GIS(Geographic Information System) is an information system to utilize geographical information for people[16]. The GIS combined with the Internet technology is referred to as Web GIS[12,15] which is able to manage and deal with the geographical data on the Internet environment. “KG21Search”, a regional information retrieval system, is referred to as a Web GIS in a broad sense.

However, there are some properties which distinguish “KG21Search” from general Web GIS’s. The regional information provided by “KG21Search” is the information which is extracted from the Web space, by analyzing structures or relationships between Web pages and keywords on the Web pages, where the Web space is regarded as a source of information. Therefore the information provided by “KG21Search” with a data mining technique, is distinguished from the information provided by a conventional Web GIS where the information is obtained through a manual operation by system administrator.

“KG21Search” is a regional information retrieval system which is available on the Internet at present. This system has a user interface to provide regional information efficiently in a close cooperation between three interfaces, namely, Map interface, Keyword interface and URL interface. For example, if a user wants to find the regional information about “Seoul” with the Web browser on “KG21Search”, the user first enters a keyword, “Seoul” in the text field. Then “KG21Search” server retrieves regional information related to the term “Seoul”, and transfers the results to the browser. The regional information is displayed on the interface of a “KG21Search” client.

As seen in Figure 1, the Global map, the left side of Map interface, shows the whole area for servicing regional information. On the Global map, the red rectangle indicates a region which is displayed on the Detailed map, and the black dots indicate Geowords related to the given keyword by user, where Geoword means a geographically significant name of locations, for example, park,

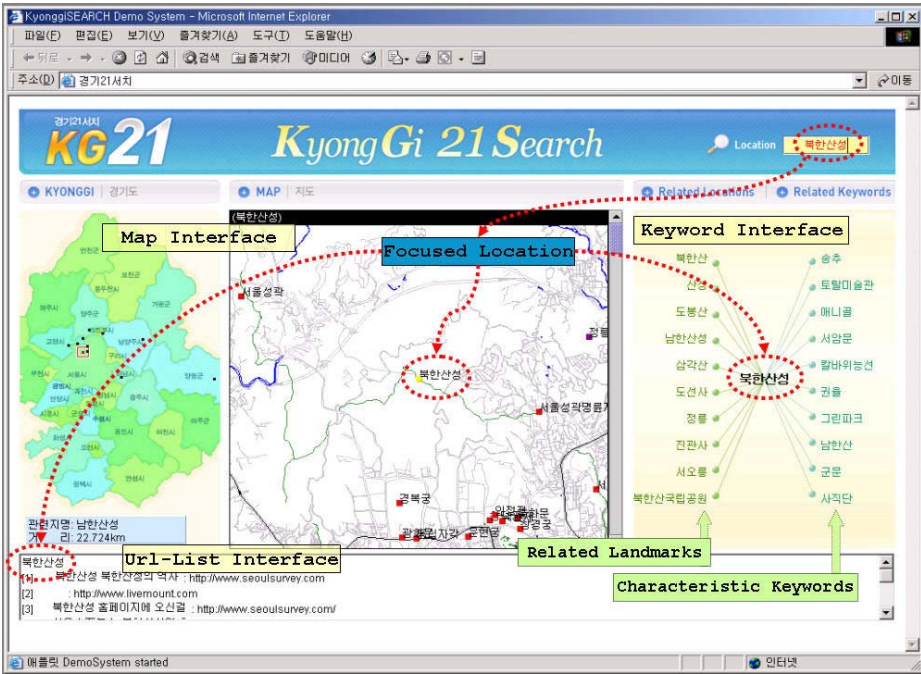


Fig. 1. KG21Search.

street, executive agency, sightseeing place, etc. A non-Geoword is not the name of locations, but terms related to the input keyword for culture and/or society. For example, those are historical events and items, social happenings, etc. The geographical distribution of the Geowords is shown on the Global map. The Detailed map, the right side of Map interface, provides detailed geographical information related to the given keyword. On the Detailed map of Figure 1, the yellow dot indicates the keyword, and the violet dots indicate Geowords related to the keyword. The Keyword interface shows the Geowords and non-Geowords highly related to the given keyword, and plays a role in a navigator for user to retrieve regional information. For example, regional information displayed on each interface can be changed when a user inputs text, handles the Map interface or selects a keyword on the Keyword interface. Finally, the URL list provides a URL list of Web pages related to the given keyword. The users can click a URL in the URL list to obtain detailed regional information over the Web browser.

3 Mobile SeoulSearch

3.1 Overview of Mobile SeoulSearch

We developed a mobile regional information retrieval system, called “Mobile SeoulSearch”, which is a mobile version of “KG21Search”. On developing “Mobile SeoulSearch”, it is necessary to maximize the mobility of mobile devices,

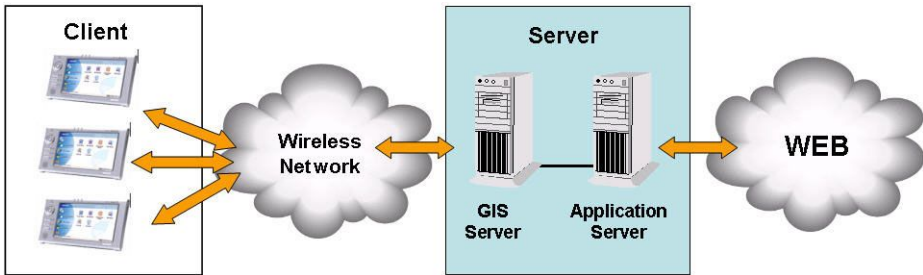


Fig. 2. Overview of Mobile SeoulSearch.

to overcome the limitations of input methods on the devices, and to improve graphic user interface on the devices, so that lots of regional information related to a mobile user's specific location, are serviced efficiently in spite of very poor user interface environment in mobile devices.

"Mobile SeoulSearch" mainly consists of a client part and a server part. The client part is a kind of PDAs equipped with the GPS, installed a developed viewer which displays lots of regional information related to a specific Geoword on a proposed graphic user interface. The server part contains two server systems, one is a GIS server, the other an Application server. The GIS server plays a role in determining the closest Geoword to a mobile user with the client, based on location information of the mobile user which is obtained from the client through wireless network. The mission of the Application server is as follows: the server receives a specific Geoword from the GIS server and retrieves "Keyword DB" on the Application server for regional information related to the specific Geoword, like a list of Geowords and non-Geowords associated with the specific Geoword, and URL information of the Geoword. This regional information is transferred to the client via the GIS server.

The details of the client and servers are presented in the subsequent sections.

3.2 Mobile SeoulSearch Server

The server part of "Mobile SeoulSearch" consists of a GIS server and an Application server. The GIS server first obtains the location of a mobile user having a mobile device equipped with the GPS through the wireless Internet. With the location data, the GIS server searches a Geoword closest to the user's present location in the spatial database and transfers the Geoword to the Application server. Figure 3 shows an example of the GIS server operating to search the closest Geoword("Gwanghwamun") if a user is traveling in the city of Seoul and carrying a PDA which has the "Mobile SeoulSearch" client software equipped with GPS.

Development of the GIS server is based on [1,2] and other methods applying R-Tree[11]. Furthermore, for maximizing the mobility of mobile devices and the speed of data transmission, we calculate a vector corresponding to the direction of mobile users, based on user's location information which is obtained in real-

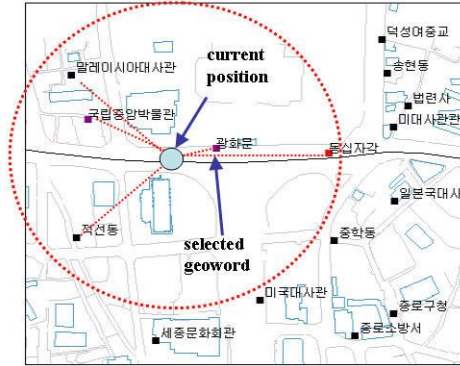


Fig. 3. An Example of GIS Server Operation.

time. Using the vector, the next location of a mobile user is calculated, and a Geoword corresponding to the next location is obtained. This Geoword is transferred to the Application server that returns regional information related to the Geoword to the GIS server. The regional information is sent to the mobile devices by the GIS server in advance, and is saved in the devices, and is displayed on the user interface of the devices by the mobile user's demand.

The Application server consists of a regional information retrieval engine, "RIRE" and a database, "Keyword DB". "RIRE" collects regional information related to a specific Geoword in a specific region, applying the methods which are presented in Section 4. "Keyword DB" has Geoword lists and non-Geoword lists related to a specific Geoword, where both lists have 10 terms, respectively.

3.3 Mobile SeoulSearch Client

"Mobile SeoulSearch" client is mobile device like PDA equipped with GPS, sends location information of mobile devices to the GIS server, and receives regional information related to the location. By the mobile user's demand, the regional information is displayed on LCD screen of the devices. However, the LCD screen has poor user interface to display, e.g., a low resolution, a small size and so on.

We propose a layered and reconfigurable graphic user interface similar to CardLayout[13] in Java, so as to display lots of regional information on LCD screen efficiently in spite of very poor user interface in mobile devices.

The interface consists of Global map view, Detailed map view, Keyword list view and URL list view. These four views are overlapped in a documentation template. Two documentation templates are usually simultaneously displayed on one window, and mobile users can select any view freely among the four views on each documentation template. Moreover, each view can be extended to full window depending on user's option(see Figure 5).

The Global map view shows the whole area for servicing regional information. On Global map view of figure 6, the whole area of Seoul and its suburb is shown. On the Global map view of Figure 6, the red rectangle indicates a region which

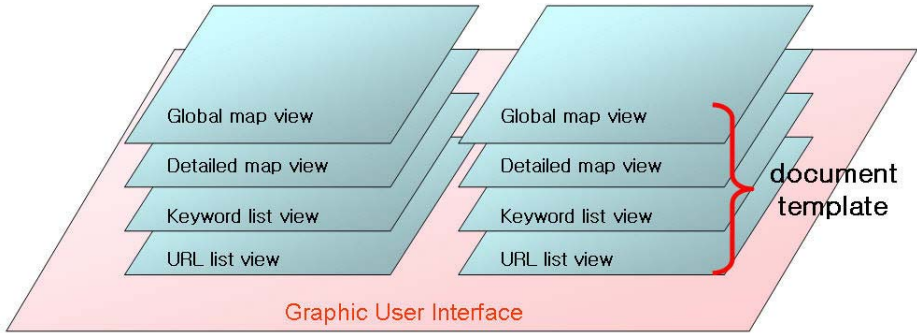


Fig. 4. The Graphic User Interface of Mobile SeoulSearch.

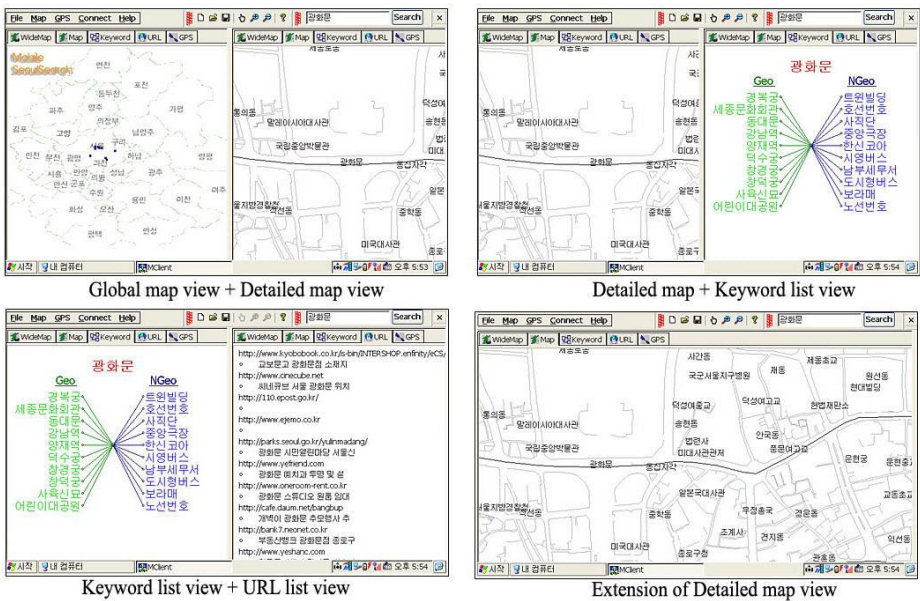


Fig. 5. An Example of User Interface Configuration.

is displayed on the Detailed map view, and the black dots indicate Geowords related to the focused Geoword which is the closest one to the current user's location. The geographical distribution of the Geowords is shown on the Global map view.

The Detailed map view provides a detailed geographical information related to the focused Geoword. On the Detailed map view of Figure 6, the yellow dot indicates the focused Geoword, and the violet dots indicate the Geowords related to the focused Geoword.

The Keyword list view provides the information of Geowords on the left side and non-Geowords on the right side, related to the focused Geoword. By clicking

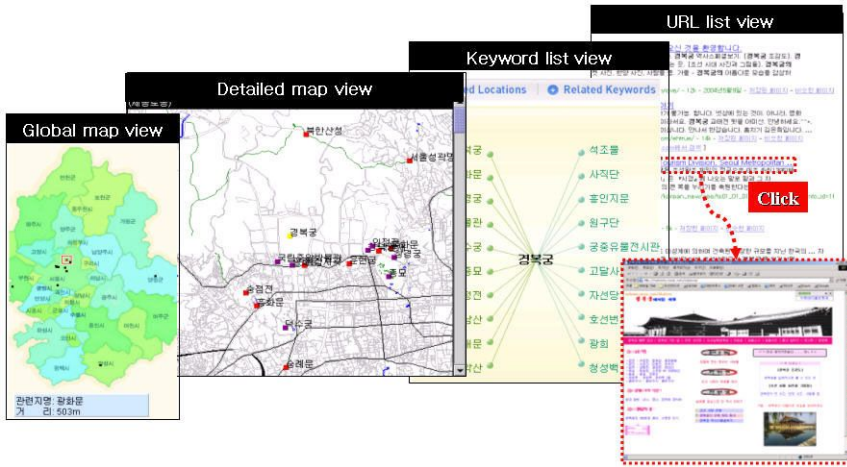


Fig. 6. The various Regional Information on an each Interface.



Fig. 7. Target Device equipped with the GPS.

any Geoword on the left side, the Geoword is sent to the GIS server, and regional information related to the Geoword is obtained, so that new regional information is displayed on the proposed graphic user interface.

Finally, the URL list view provides a URL list of Web pages related to the focused Geoword. The users can click a URL in the URL list to obtain the detailed regional information over the Web browser.

3.4 System Detailed

Implementation of GIS server and Application server is performed on the system, Xeon 2.4G 4-CPU, 4G RAM, RAID5, Window 2000 with develop platform, Microsoft Visual C++ 6.0 platform.

For the mobile client, we use a device, Intel PXA255 400Mhz CPU, 128RAM with Windows CE 4.1 OS as the target device, and for development toolkit, Embedded Visual C++ 4.0 SP2 is used.

4 Extracting Regional Information from Web

4.1 Gathering Web Pages Related to a Specific Region

To achieve the service mentioned above, “Mobile SeoulSearch” regards the Web space as a source of regional information related to a specific region, and gathers the Web pages related to a specific region. Later, regional information related to a specific region is extracted by analyzing these pages, and the extracted regional information is displayed on the Global map view, the Detailed map view, the Keyword list view and the URL list view.

Figure 7 shows the configuration diagram of a system to extract regional information related to a specific region.

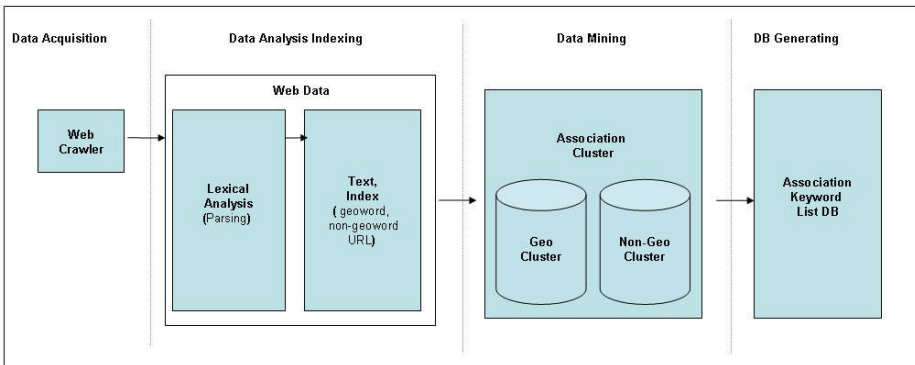


Fig. 8. Process for Extracting regional Information.

Generally, classifying Web pages, e.g., gathering Web pages related to a specific region, applies the techniques of Automatic Classification [9,10]. However, it is difficult to apply the Automatic Classification since it requires an appropriate standard for classification such as a thesaurus or word distribution pattern. Even if there is a thesaurus related to the specific region, it still takes too much time and effort to classify the whole Web pages. Therefore in this paper, we apply the following method described below to gather Web pages related to a specific region, which is similar to that used in “KG21Search”.

For example, suppose that we are gathering Web pages related to the region of “Seoul”, the capital of Korea. Generally, querying “Seoul” to the search engine, it will give users a URL list of Web pages which are judged to be highly associated with the query. However, those Web pages have the following problems: 1) They do not guarantee that the set of Web pages including the term “Seoul” also

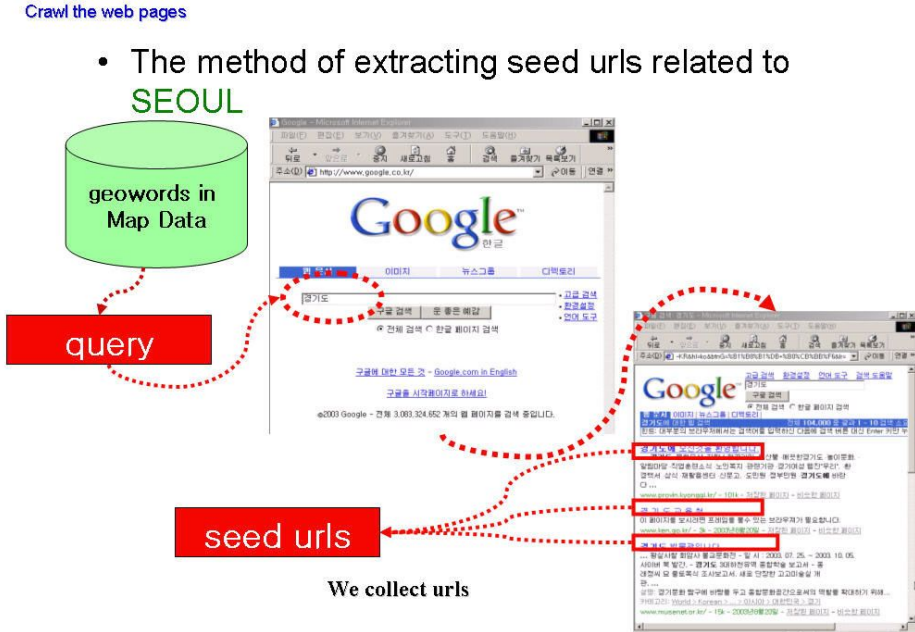


Fig. 9. Collection of Web Data utilizing the Search Engine.

include all of the Geowords related to “Seoul”. 2) They do not have some Web pages related to the region “Seoul” in case that the Web pages do not have the term “Seoul”.

Therefore, it is required to gather the Web pages by querying the terms related to the region “Seoul” as well as the term “Seoul”. However it is still difficult to obtain the terms related to the region “Seoul” because a thesaurus for gathering terms related to the specific region is needed.

Thus, vector maps of some regions are utilized to collect all of the Geowords included in a specific region in this paper because those maps are a kind of formatted data and contain all Geowords of a specific region. The names of places in a vector map of a specific region, such as executive agency name, park, street or station, are regarded as the Geowords related to the specific region. These Geowords are used for gathering Web pages related to the specific region.

On the map of the region “Seoul”, there are 1,434 words regarded as Geowords in our research. Then these words are queried over “Google”, “Naver” and “Empas” which are search engines mostly used in Korea, so that about 180,000 Web pages are gathered as those related to the region “Seoul”.

4.2 Extracting Indexes Related to a Specific Region

In this Section, we discuss the method of extracting indexes related to a specific region from the gathered Web pages, where this process is a pre-process for calculating associativity between extracted indexes.

“Mobile SeoulSearch” utilizes existing search engines to extract indexes related to a specific region, as “KG21Search” does the same. Generally, when a user gives a term to the Web search engine as a query, the search engine commonly returns a URL list of Web pages related to the term and displays an abstract of each Web page. This abstract usually includes the query. The process of extracting the indexes utilizes this common property of search engines as follows: If there is the name of a specific region in the abstracts of the result page, then the term is judged to be related to the specific region. Depending on situations, it is necessary to adjust the number of appearance of the name in order to make the extraction more accurate. Through this method, a set of indexes related to a specific region can be extracted out of all terms derived from the gathered Web pages. Figure 10 shows the process to decide whether a term in the gathered Web pages is related to a specific region.



Fig. 10. An Example of the Result Page of a Query.

An actual processing of extracting the indexes is as follows: First of all, to create a candidate set of terms for indexes, HAM[13], a Korean Morphemes analyzer is applied to the gathered Web pages. 372,509 terms are obtained from approximately 180,000 Web pages. And then the 372,509 terms are given to the search engines as queries to create the set of indexes related to the region “Seoul”. Finally 50,172 terms related to the region “Seoul” are extracted.

Extract the words related to specific location

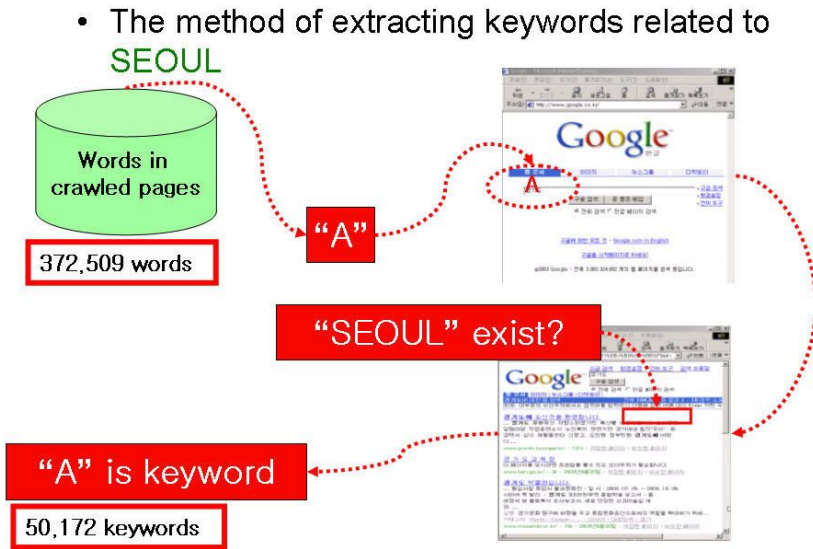


Fig. 11. Extracting a Word Related to a Certain Geoword.

Table 1 shows 10 high-ranked terms in occurrence frequency from the set of 372,509 words gathered through the HAM at the left side, and the set of indexes related to the region “Seoul” at the right side, respectively. The occurrence frequency indicates how many times a specific term is occurred in the whole gathered Web pages. The terms such as “Selection” and “Information” are not decided to be indexes related to the region “Seoul”, since they are not co-occurred with the term “Seoul” in the result page. In conclusion, the method of extracting indexes related to a specific region through a search engine gives relatively better performance.

Table 1. Comparing terms obtained through HAM and indexes related to the region “Seoul”.

terms obtained through HAM		indexes related to the region “Seoul”	
Terms	Occurrence Frequency	Terms	Occurrence Frequency
Kyonggi-do	70,479	Kyonggi-do	70,479
Select	55,357	Gwangyuk-city	34,251
Homepage	51,908	Gapyeonggun	18,471
Modify	49,011	Suwon-city	16,078
Guide	44,085	Bucheon-city	15,779
Register	43,732	Seongnam-city	15,419
Mail	43,248	Management	15,018
Information	43,129	Goyang-city	14,491
Naver	42,454	Ahnyang-city	13,645
Shopping	42,021	Yongin-city	13,560

4.3 Elimination of Unnecessary Terms in Mobile SeoulSearch

From the preceding discussion in Section 4.2, indexes related to a specific region have been gathered. However, there still exists some ordinary terms which cannot be verified whether to have relationship with a specific region in the set of extracted indexes. The terms frequently occurred in the general Web pages are regarded as ordinary terms in “Mobile SeoulSearch”. Those terms are of little worth in providing regional information related to a specific region.

Therefore, ordinary terms are regarded as unnecessary terms and need to be eliminated from the set of indexes related to a specific region to provide regional information more efficiently in this paper. A set of ordinary terms can be created by inspecting the frequency of occurrence of all terms in the whole Web pages and regarding frequently occurred terms as the ordinary terms. However, it costs too much to inspect the whole Web pages in Web space, so that we select some samples of Web pages and inspect the frequency of occurrence of the terms in them. Most of Web sites are biased in their contents, and generally the terms occurred in those Web sites are closely related to its content. Thus those Web sites are not suitable for gathering the set of ordinary terms. On the other hand, the Web sites for News have various subjects of contents and use the standard language, so that they are suitable for samples to obtain the ordinary terms.

Therefore a Web site for News is selected and inspected to create the set of the ordinary terms. In practice, we gathered the Web pages of “Chosun.com”, a Web site of Korean Newspaper, and gathered about 56,000 pages. Using HAM, approximately 500,000 terms are extracted. Among these terms, the ordinary terms are determined depending on their frequency of occurrence, e.g., up to 5%, 10% or 20% from above. However the Geowords in the vector map have to be excluded from the set of ordinary terms. For example, the terms “Kyounggi-do” and “Suwon-city” in the vector map are also in the Web pages of “Chosun.com” with the high frequency of occurrence, 2% from above. Thus these two terms are excluded from the set of the ordinary terms.

4.4 Calculating Associativity Between Extracted Indexes

In general, the methods applying association rule and association cluster are used to calculate associativity between extracted indexes.

Indexes handled by “Mobile SeoulSearch” are cultural, geographical or historical terms which are usually biased in their distribution in the Web space. The frequencies of co-occurrence of terms as well as co-occurrence of terms play an important role in calculating associativity between the extracted indexes. The method applying association cluster to co-occurrence and the frequency of co-occurrence is more efficient in calculating associativity between extracted indexes than that applying association rules only to co-occurrence of terms. Thus, the method applying association cluster is adopted in “Mobile SeoulSearch” to calculate associativity between indexes.

The process of calculating associativity between indexes is as follows: First, measures the occurrence frequency of index w_i in Web page D_j to create matrix

m_{ij} . Then, create a transpose-matrix of m_{ij} , m_{ij}^t , to obtain the association matrix between the indexes by using product of the two matrixes, $s = m_{ij} \times m_{ij}^t$, where the elements of matrix s mean associativity of the co-occurrence frequency of both indexes in the same document. This process is summarized as follows:

① Measure the occurrence frequency of keyword w_i in document d_j to create matrix m_{ij} .

② Create transpose matrix $m^t = m_{ij}$.

③ Calculate associativity $s = m_{ij} \times m_{ij}^t$,

$$\begin{array}{cccccc}
 & d_1 & d_2 & \cdots & d_k & & w_1 & w_2 & \cdots & w_l & & w_1 & w_2 & \cdots & w_l \\
 w_1 & \cdot & \cdot & \cdots & \cdot & & d_1 & \cdot & \cdot & \cdots & \cdot & & w_1 & \cdot & \cdot & \cdots & \cdot \\
 w_2 & \cdot & \cdot & \cdots & \cdot & & d_2 & \cdot & \cdot & \cdots & \cdot & & w_2 & \cdot & \cdot & \cdots & \cdot \\
 \cdots & \cdot & \cdot & \cdots & \cdot & & \cdots & \cdot & \cdot & \cdots & \cdot & & \cdots & \cdot & \cdot & \cdots & \cdot \\
 w_l & \cdot & \cdot & \cdots & \cdot & & d_k & \cdot & \cdot & \cdots & \cdot & & w_l & \cdot & \cdot & \cdots & \cdot
 \end{array} =$$

However, taking the product of the two matrixes requires high-cost operation. To solve this problem, the property that matrix m_{ij} is very sparse is applied in this paper, so that the time taken to calculate the production is reduced greatly.

5 Conclusion and Future Work

In this paper, we presented “Mobile SeoulSearch” developed as a regional information retrieval system for mobile environment. This system searches the nearest Geoword to a mobile user’s location which is detected in real-time, and provides regional information related to the Geoword, where the regional information is derived from the Web space. “Mobile SeoulSearch” is said to be one of the most fitted mobile regional information retrieval system that utilizes the features of mobile environment efficiently, and coping with the limitation of user interface in mobile devices.

For developing “Mobile SeoulSearch”, several methods are proposed: Firstly, we proposed a method that regional information related to a mobile user’s specific location is provided in real-time to mobile devices, which is based on user’s location information that is obtained automatically by means of calculating a vector corresponding to the direction of a mobile user. Using this method, the mobility of mobile devices is maximized, and the limitation of input methods on the devices is overcome.

Secondly, we proposed a layered and reconfigurable user interface similar to CardLayout in Java. With the user interface, lots of regional information related to a Geoword, are displayed efficiently in spite of very poor user interface environment in mobile devices.

Finally, utilizing the existing Web search engines and the formatted data of vector maps, we proposed a method for gathering Web pages and for extracting indexes related to a specific region at high speed. Owing to this method, “Mobile SeoulSearch” can provide mobile users with regional information related to a specific Geoword effectively.

The efficiency of the above methods are verified through the development of “Mobile SeoulSearch”.

The future work includes development of a personalized mobile regional information retrieval system based on the history and/or patterns of mobile user's behavior.

Acknowledgements

This research was supported by IRC (Internet Information Retrieval Research Center) in Hankuk Aviation University. IRC is a Kyounggi-Province Regional Research Center designated by Korea Science and Engineering Foundation and Ministry of Science & Technology.

References

1. R. Lee, H. Shiina, H. Takakura, Y.J. Kwon, and Y. Kambayashi, "Map-based Range Query Processing for Geographic Web Search Systems", *Digital Cities 3 Workshop: local information and communication infrastructures experiences and challenges*, pp. 1-10, Amsterdam, The Netherlands, September 18 and 19, 2003.
2. R. Lee, H. Shiina, H. Takakura, Y.J. Kwon, Y. Kambayashi, "Optimization of Geographic Area to a Web Page for Two-Dimensional Range Query Processing", *Web Information Systems Engineering Workshops*, 2003. Proceedings. Fourth International Conference on, pp. 1-9, 13 Dec. 2003
3. Y. Ishikawa, Y. Tsukamoto, H. Kitagawa, "Implementation and evaluation of an adaptive neighborhood information retrieval system for mobile users" *Web Information Systems Engineering Workshops*, 2003. Proceedings. Fourth International Conference on, pp. 25 - 33, 13 Dec. 2003
4. Jung-Hoon Jang, Do-Hyoung Kim, Ryong Lee, Yahiko Kambayashi, Yong-Jin Kwon, "An Optimizing Method for Generating Index Association by Using Property of Sparse Matrix", *Proc. of International Conference on Internet Information Retrieval 2003*, pp. 229-232, Koyang City, Korea, October 30, 2003.
5. Jung-Hoon Jang, Ryong Lee, Yahiko Kambayashi, Yong-Jin Kwon, "Implementation of "Kyonggi21Search" combining GIS with The Web:Optimization of Index Association", *Proc. of The 30th KISS Fall Conference, VOLUME 30, NUMER 2*, pp. 79-81, Seoul, Korea, October 24 and 25, 2003.
6. Jung-Hoon Jang, Do-Hyung Kim, Ryong Lee, Yahiko Kambayashi, Yong-Jin Kwon, "Implementation of "Kyonggi21 Search" combining GIS with Web : index abstraction, index association, user interface", *2003 Symposium on Regional IT Industry Growth*, pp. 19-23, Daejun, Korea, June 27, 2003.
7. R. Lee, Y. Tezuka, N. Yamada, H. Takakura, Y. Kambayashi, "KyotoSEARCH: A Concept-based GeographicWeb Search Engine," In *proceedings of 2002 IRC International Conference on Internet Information Retrieval*, pp. 119-126, Koyang, Korea, Nov. 2002. [8] Yae-Kwan Yun, Yeom-Seung Jang, Gi-Jun Han, "Location Based Service for Mobile GIS", *Proc. of the SIGDB VOL. 18 NO. 01 pp. 0003 0015 2002.03*
8. Yae-Kwan Yun, Yeom-Seung Jang, Gi-Jun Han, "Location Based Service for Mobile GIS", *Proc. of the SIGDB VOL. 18 NO. 01 pp. 0003 0015 2002.03*
9. Hye-Ju Eun, Yan Ha, Yong-Sung Kim, "An Algorithm of Documents Classification and Query Extension using Fuzzy Function", *Proc. of The KISS, VOLUME 28, NUMER 3*, pp. 272-284, Seoul, Korea, March, 2001.

10. Soo-Jung Ko, JunnHyeog Choi, Jung-Hyen Lee, "Optimization of Associative Word Knowledge Base using Apriori-Genetic Algorithm Proc. of The KISS, VOLUME 28, NUMER 08, pp. 560 569 Seoul, Korea, August, 2001.
11. G.Antonin, R-TREE : A Dynamic Index Structure for Spatial Searching, In Proceeding of ACM SIGMOD, pages,45-57,1984
12. Bong-Heui Hong, Sang-Ho Mun, Un-Mo Sung, "integration technology of GIS and Internet", Proc. of the SIGDB, VOL. 12 NO. 03 pp. 0097 0115, 1996.08
13. Korean morphological analyzer,
<http://nlp.kookmin.ac.kr/HAM/kor/ham-intr.html>
14. CardLayout,<http://java.sun.com/developer/technicalArticles/GUI/>
15. e-GIS, <http://www.e-gis.or.kr/>
16. GIS, <http://www.gis.com/whatisgis/index.html>

A Novel Indexing Method for Digital Video Contents Using a 3-Dimensional City Map

Yukiko Sato¹ and Yoshifumi Masunaga²

¹ Division of Mathematics and Computer Science
Graduate School of Humanities and Sciences
yukiko@dblab.is.ocha.ac.jp

² Department of Information Science
Faculty of Science, Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku, 112-8610 Tokyo, Japan
masunaga@is.ocha.ac.jp

Abstract. This paper presents a novel indexing method for digital video contents. The method automatically identifies city buildings captured by digital video camera. This is done by extracting the objects from candidate objects using GPS location-and-time data of the video shooter, video camera posture data taken by a Gyro attached to the camera, and a 3-dimensional city map stored in a database. The automatic calculation identifies the start and end video frames for each building object captured in a video unit. An index is created to refer to the set of all video units in which a specified building is really captured. A concrete experiment implementing the proposed algorithm in Ginza Area, Tokyo demonstrated that the algorithm works as designed.

1 Introduction

Recent progress in electronics and precision instrument technology makes it possible for video cameras to be very compact and for the life expectancy of batteries to be longer; both of these developments enable people to take much longer “home videos”. As a matter of course, huge amounts of video data are accumulated in homes or businesses. In addition, dramatic progress in video data handling techniques such as MPEG compression standards as well as data management technology for large amounts of stream data has enabled us to build systems for large video databases.

Of course, if a video database system cannot provide an effective video retrieval function, it is doubtlessly of little worth. Therefore, the development of an accurate indexing technique for video databases has been recognized as one of the most important research issues in this field [1] [2]. In the early stage, intensive research and development of shot boundary detection was done to evaluate how accurately a system could automatically detect camera shot boundaries in video contents [3]. Notice that a video is composed as a video clip, a video clip is a consecutive sequence of video scenes (or stories) which are the logical units of a video, and a video scene is a consecutive sequence of shots which are the physical units of a video, where a shot is a sequence of video frames captured between a single record and stop camera operation. Research on developing automatic detection algorithms for video scenes from a

video clip has also received much attention. However, these approaches are considered to be the lower level of video processing in comparison to content-based approaches.

In order to create indices or annotations on a video clip or a video scene to refer to its contents, most of the previous research has adopted “content-based” audiovisual feature extraction techniques such as an image understanding technique, an object extraction and tracing technique, an audio recognition technique, a speaker recognition technique, a character recognition technique, or a camera-work information extraction technique. A content-based approach is attractive for several reasons: it could provide an effective query language for video retrieval, it could realize a similarity search among videos, or it could break down a video clip into a sequence of key frames for browsing purposes. However, none of these techniques will be easy to achieve.

In order to overcome such difficulties and achieve a higher accuracy for video indexing, recent studies have focused intensively on semantic approaches. For example, to provide concept-level access to video, an approach based on the use of knowledge models to build domain specific video information systems was proposed [4]. The use of rule-based domain-specific knowledge was examined to build a decision-tree based video classification system taking a sport video clip example [5]. However, it is difficult to extend the results of these approaches to other domains because the domain-specific knowledge is different among application domains.

In this paper, we propose a novel indexing method of content-based video indexing in that our approach calculates directly what objects are captured in each video frame of a video clip. The following describes an environment where our approach works well:

- When a GPS (Global Positioning System) is used to capture a video shooter’s position and shooting time data;
- When a Gyro sensor is used to capture a video camera’s posture data; and
- When a 3-dimensional city map is used to retrieve the location and the height information of buildings in a city.

In other words, we presume that the video shooter is walking on a street of a city and is equipped with GPS and a Gyro so that the sequence of video frames is collected along with the shooter’s position data and camera’s posture data in a wearable computer. These data are processed using 3-dimensional map data in either on-line or off-line settings to create an index to the video clip based on identifications of the buildings captured in the video clip.

The use of a GPS and a map has received attention recently for video indexing. For example, the idea of the work done by Ueda *et al.* [6] is quite similar to ours in that both approaches use a map as well as GPS to index video contents. However, these two works are completely different in that Ueda *et al.* used a 2-dimensional map while we used a 3-dimensional map, which caused this essential difference: Because 2-dimensional maps cannot provide the height information of a building or a landmark object, they cannot compute what building objects are captured in a video frame. In contrast, this computation can be done perfectly in our approach because we use a 3-dimensional city map. Therefore, the previous researchers aimed to develop a user support system that can aid a video shooter to think back on his/her past behavior

by showing the video scene of the shooting area where the shooter was located at a specified time. Our approach can answer a query such as “Show me a scene in which Ginza Mitukoshi Department Store appears more than 10 seconds” while theirs cannot. However, their system can answer a query such as “Show me a scene when I was in the Ginza Mitukoshi Department Store area.” (Notice that the scene may not include the Ginza Mitsukoshi Department Store.) A similar work to [6] for developing a context-based video retrieval system for life-log applications was reported in [7].

2 Basic Idea and Approach

2.1 Concept of an Automatic Extraction and Indexing System

Fig. 1 represents the concept of the system under development in our project. We use GPS to obtain the video shooter’s global position and a Gyro sensor to obtain the video camera’s posture data. The shooter is equipped with a wearable computer to process GPS data, Gyro data, camera lens angle data, and 3-dimensional building object data so that the building objects captured in a video frame can be identified. We use a digital video camera (Sony DCR-PC1) with lens angles of 44° (width) by 35° (height) and a 3-dimensional map, DiaMap, created by the Mitsubishi Corporation. In order to obtain the primary building names in a city, we use a 2-dimensional map named Zmap-Town II of Zenrin.

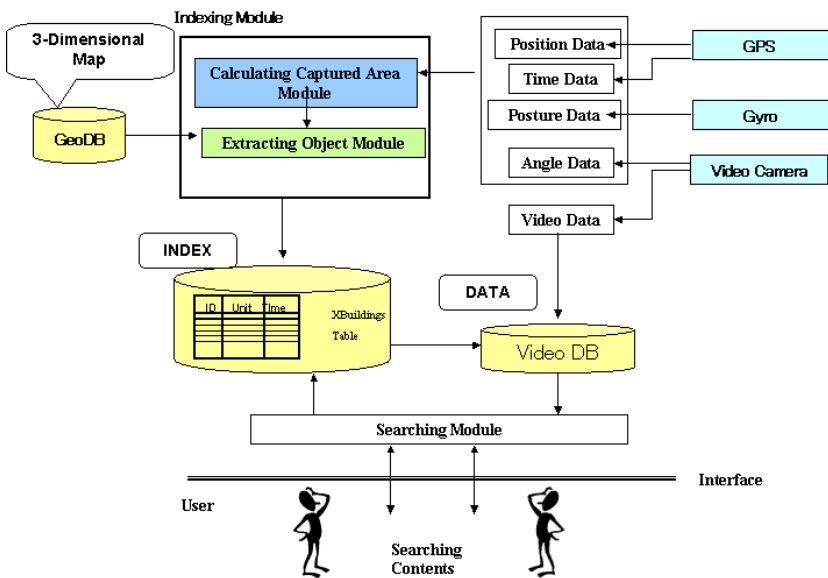


Fig. 1. Concept of an Automatic Extraction and Indexing System.

By processing those data comprehensively in the manner mentioned below, an INDEX table named “XBuildings” is created in the video database. The XBuildings table has attributes such as “BuildingObjectID” and “BuildingName”, as well as the

start and end frame numbers of a video unit in which the building with a specified BuildingObjectID value is captured continuously. Since a BuildingObjectID is simply a symbol string, it is converted to a specific building name using Digital Map 2500 of the Geographical Survey Institute.

2.2 Using a 3-Dimensional City Map

Fig. 2 shows a sample of a 3-dimensional map scene around Ginza Marion Mall. A geographic information system named ArcView 3.2 of ESRI is used to display the image. In the picture the square pyramid shows the spatial area which a video camera could capture, if the video shooter were located at the apex of a square pyramid. The buildings inside or intersecting with this square pyramid and not hidden by the buildings in front of them are the buildings which are actually captured by this video camera at a certain point in time.

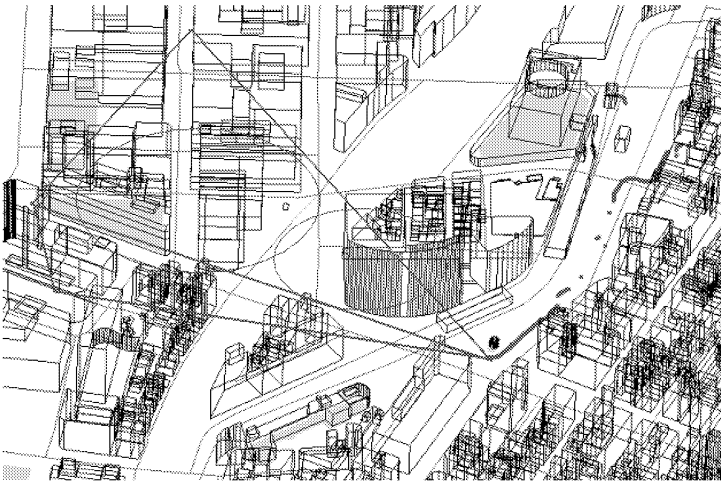


Fig. 2. 3-Dimensional Map for the use of Automatic Extraction of Building Objects.

2.3 Video Indexing

A video shooter walks on a street shifting the video camera up, down, right and left. Suppose that $u_{v,o,i}$ is the i -th portion of a video clip v where a building object o is continuously captured from the beginning video frame fs_i up to the ending video frame fe_i . Then the video unit $u_{v,o,i}$ is a quadruple defined by $u_{v,o,i}=(v, o, fs_i, fe_i)$. Fig. 3 shows a sample relation between building objects and video units where object o_1 is captured twice. To reflect the relation, index table XBuilding includes at least two tuples; $(o_1, u_{v,o,1})$ and $(o_1, u_{v,o,3})$. Therefore, the system can directly answer a query like “Show me a video scene where building object o is captured.” Since a video is captured at the frame rate of 30, and GPS can supply the start time of a shot, we can compute the captured time so that the system can answer a query like “Show me a video scene where the building object o has been captured more than 10 seconds.”

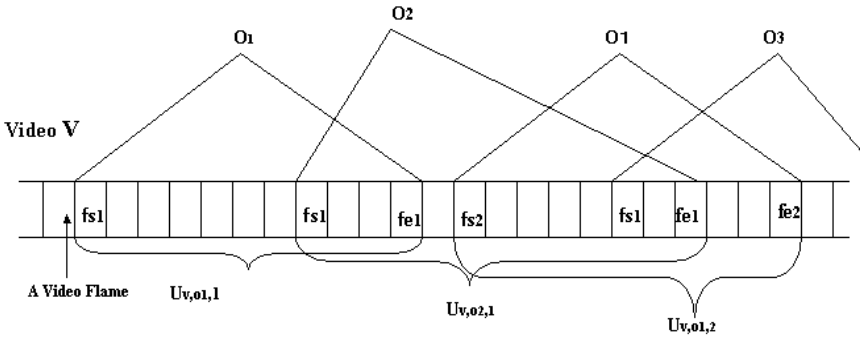


Fig. 3. Relation between Building Objects and Units.

3 Automatic Extraction Algorithm

3.1 Two-Step Approach

In this section, we show an algorithm that can automatically calculate all the 3-dimensional building objects which are really captured in a video frame. As the Indexing Module of Fig. 1 indicates, the algorithm consists of the following two steps: In order to identify a set of candidate building objects to be captured in a video frame, the first step projects a square pyramid in the 3-dimensional space captured by a video camera into a triangle on the 2-dimensional X-Y plane which represents the ground. It is clear that the candidate building objects are buildings inside or intersecting with the triangle. The second step compares the heights of candidate buildings so that the building objects that are not hidden by buildings in front of them are really extracted. Notice that the necessary building height data are provided by the 3-dimensional city map.

(1) Identification of a triangle on the 2-dimensional X-Y Plane

As mentioned above, the first step extracts the candidate building objects. As shown in Fig. 4, the square pyramid P represents the 3-dimensional subspace which is captured by a video camera at time τ with respect to view length L . L represents the maximum distance from the video camera; all buildings within this distance are checked for visibility in the video frame. The shadowed area represents a triangle T obtained by projecting P onto the 2-dimensional X-Y plane, which represents the ground. Considering the future use of the zoom function of video cameras, a yaw angle (α) and a roll angle (β) of a Gyro sensor are configured to the center of the width degree and the height degree, respectively. View length L is set small in a city where the buildings are crowded, while L is set large in a country where the buildings are sparse. Since our experiment was done in Ginza, a central part of Tokyo, we set $L = 80\text{m}$. The width angle of the video camera in use was set at 44° . Let l represent the projected view length on the 2-dimensional X-Y plane, which is calculated by $l = L \times \cos(\beta)$. Then, the triangle T is characterized by three apexes; $T' = (0, 0)$, $T_0 = (l \times \cos(\alpha - 22), l \times \sin(\alpha - 22))$, and $T_{44} = (l \times \cos(\alpha + 22), l \times \sin(\alpha + 22))$. Obviously the captured building objects should reside in T or intersect with T .

(2) Extraction of Captured Building Objects

The second step extracts the set of building objects that are actually captured in a video frame at time τ . This extraction is performed using the height information about buildings supplied by DiaMap, a 3-dimensional city map provided by the Mitsubishi Corporation. Considering the video camera's vertical angle (35°), roll angle (β), and shooter's height (h), the building objects captured by the video camera must be inside the square pyramid P which is characterized by five apexes; $P_1=(0, 0, h)$, $P_2=(l \times \cos(\alpha-22), l \times \sin(\alpha-22), L \times \sin(\beta+17.5)+h)$, $P_3=(l \times \cos(\alpha+22), l \times \sin(\alpha+22), L \times \sin(\beta+17.5)+h)$, $P_4=(l \times \cos(\alpha+22), l \times \sin(\alpha+22), L \times \sin(\beta-17.5)+h)$, and $P_5=(l \times \cos(\alpha-22), l \times \sin(\alpha-22), L \times \sin(\beta-17.5)+h)$. A candidate building can be really captured by the video camera if its part is certainly "visible" from P_1 , the video camera shooter's position. Now the problem is how to decide or calculate which building objects are really captured or not. In the next section we will show an algorithm that can extract them automatically.

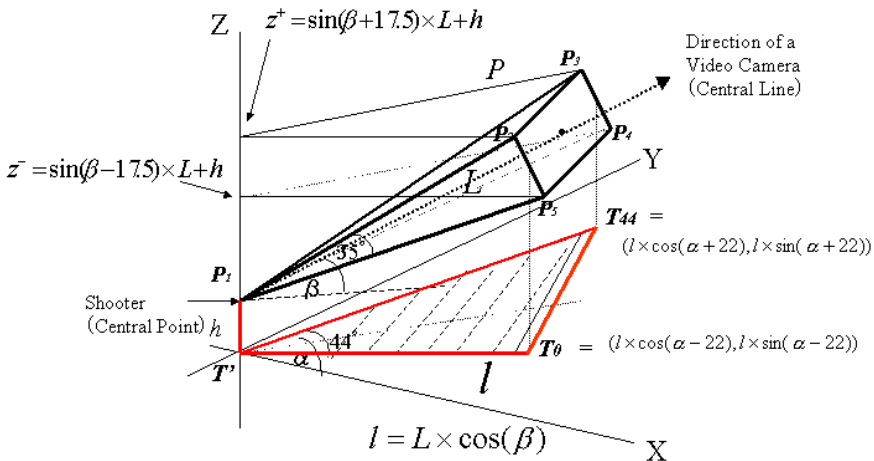


Fig. 4. Projection of a 3-Dimensional Video Captured Sub-Space on X-Y Plane.

3.2 Extraction of Really Captured Building Objects

3.2.1 Variables and Parameters

Fig. 5 shows an outline of an automatic extraction algorithm of captured building objects. The algorithm includes several variables and parameters for building object o :

- Center of gravity coordinates of the base of a building object: $(o.x, o.y)$
- Height of building object: $o.height$
- Distance of center of gravity of building object o from the shooting point: $o.r$
- Flag whether a building object o is visible (=on) or not (=off) from a shooting point: $o.visible$

3.2.2 Extraction of Candidate Building Objects on 2-Dimensional Map

In the first part of the algorithm, candidate building objects to be captured by a video camera are calculated using T which is obtained by projecting P onto the X-Y plane as mentioned in Section 3.1. From the horizontal angle (44°) of a camera lens in use and the projected view length l , which is calculated by $l = L \times \cos(\beta)$, the building objects captured by the video camera must be inside or intersect with T .

Fig. 5 shows the algorithm that was implemented in our research. We first notice a line $T'T_0$ of T and put the building objects in array $a[i][]$ whose bottoms intersect with the line. The intersection is decided using a feature selection function of Arc-View3.2, a geographic information system of ESRI under use. By shifting the line, defined by $T'T_i$ ($T_i = (l \times \cos(\alpha - 22 + i), l \times \sin(\alpha - 22 + i))$), by one degree ($0 \leq i \leq 44$) from $T'T_0$ to $T'T_{44}$, all the components of array $a[]$ $= \{a[i][] \mid i=0, \dots, 44\}$ are prepared.

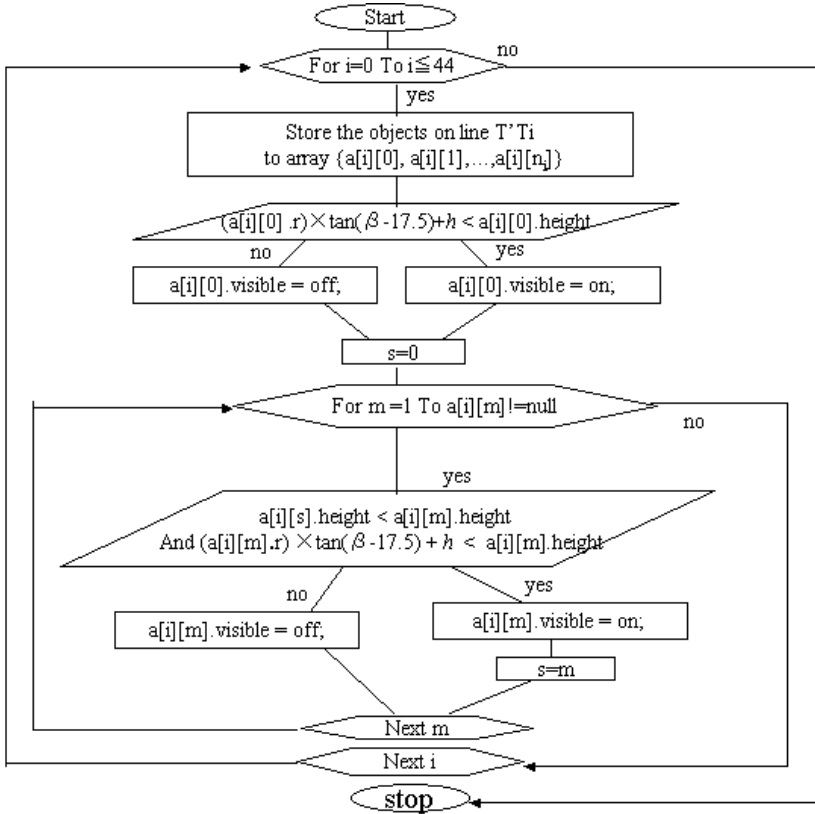


Fig. 5. Outline of an Automatic Extraction Algorithm of Building Objects.

Because of the feature selection function we used, the components of the array $a[i][]$ are not ordered by the length from the shooter point to building objects in ascending or descending order. In order to simplify the object extraction program mentioned in the next section, we assume that the components of the array are sorted in ascending order by pre-processing.

3.2.3 Object Extraction Using Building Height Information

Using the array $a[i][]$ and building height information provided by DiaMap, we can extract the building objects really captured in a video frame at time τ .

First, as it is shown in Fig. 5, in the outer loop, the line $T'T_i$ is defined for each i ($0 \leq i \leq 44$). For each i , store all the building objects on the line $T'T_i$ in the array $\{a[i][0], a[i][1], \dots, a[i][n_i]\}$, where n_i is defined in the sense that $a[i][n_i]$ is not null for all $n \leq n_i$ but $a[i][n_i+1]$ becomes null. In other words, the number of objects that intersect with line $T'T_i$ is n_i+1 . Such a number exists because we assumed that the finite view length is L and that the components of an array are sorted in ascending order. Check the height of the first element on the line $T'T_i$ (the front-most building object to the video camera shooter); $a[i][0].visible$ is set “on” if the height of the building object ($a[i][0].height$) is higher than the bottom of P ($(a[i][0].r) \times \tan(\beta-17.5) + h$) at this place.

Second, in the inner loop, the following comparison is executed as long as $a[i][m]$ is not null: Set $a[i][m].visible$ be “on” and set $s=m$ if $a[i][s].height < a[i][m].height$ and $((a[i][m].r) \times \tan(\beta-17.5) + h) < a[i][m].height$. Otherwise, set $a[i][m].visible$ be “off”. It is clear that all the elements of array $a[i][]$ whose $o.visible$ values are “on” are the building objects that are really captured by the video camera.

3.2.4 No False Dismissal

Since parameter “ i ” takes its value from 0 to 44 by “1” (degree), in principle there is a danger that there might be a building that exists between two consecutive lines. In other words, there might be a candidate building that cannot be detected by this algorithm. However, since we set $L=80\text{m}$, the distance between two consecutive lines at 80m away from the shooter’s position is calculated as $2\pi L/360 \approx 1.4\text{m}$, and since buildings in the real world are more than 1.4m in width, we can conclude that there are no buildings which should have been captured but were missed. Therefore, there is no false dismissal in our setting.

4 Experiment and Verification

4.1 An Experiment

We implemented the proposed algorithm by Avenue, a system development programming language of ArcView3.2, on a Windows XP machine. To verify the validity of the implemented algorithm, the following experiment was done:

- (1) Ginza was selected to test our algorithm because it is known as one of the most crowded building areas in Tokyo.
- (2) A Windows XP machine was used to collect GPS data and Gyro sensor data. A GPS was attached to a video shooter, and a Gyro Sensor was attached to a video camera.
- (3) Video data and GPS and Gyro sensor data were synchronized.
- (4) The algorithm was run to extract building objects that are really captured.
- (5) Comparison was done manually in order to verify the effectiveness of the proposed algorithm.

The dotted line in Fig. 6 shows the trajectory of the video shooter or video camera position captured by GPS. The shooter started around Sony Building toward the Prin-temps-Ginza Building. The video was about 10 minutes long and consisted of one video shot.

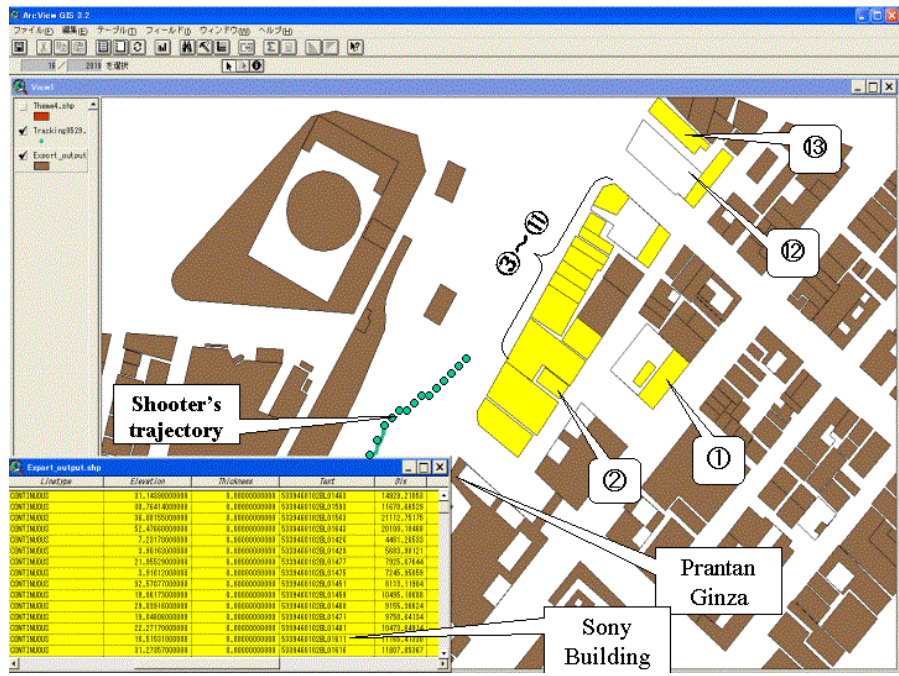


Fig. 6. Trajectory of a Video Camera and Building Objects Automatically Extracted.

4.2 Verification by Comparison

Verification was done in a following manner: We selected a time τ to compare the set of building objects captured by the real video frame at τ and the set of buildings computed by the algorithm to be captured in that frame. Notice that time τ was carefully selected so that the verification effects had universality in terms of the density of building objects, the difference among building heights, and the distance of the video camera to building objects.

The objects colored gray in Fig. 6 are building objects extracted by the calculation. There are non-extracted building objects in the background because there are high building objects in front of them. In this experiment, a total of 15 objects were extracted. The table in Fig. 6 lists the set of all extracted building objects with attributes such as a building object ID, the center-of-gravity coordinates of a building object, and the height of a building.

Now, Fig. 7 shows a real video frame at time τ . By comparing Fig. 6 and Fig. 7, we found that two buildings were missing in Fig. 7, i.e. only 13 buildings in Fig. 6 had their counterparts in Fig. 7. It turned out that the building object in the closest fore-

ground didn't exist because of construction, but the experiment certified for the others that the captured objects in a real video frame had been extracted by our method. Using a 3-dimensional map, we confirmed that the building objects that were captured in the real video were extracted and stored to the video database correctly, and objects that were hidden by neighboring buildings were not extracted. Thus, the effectiveness of the algorithm proposed in this paper was verified.

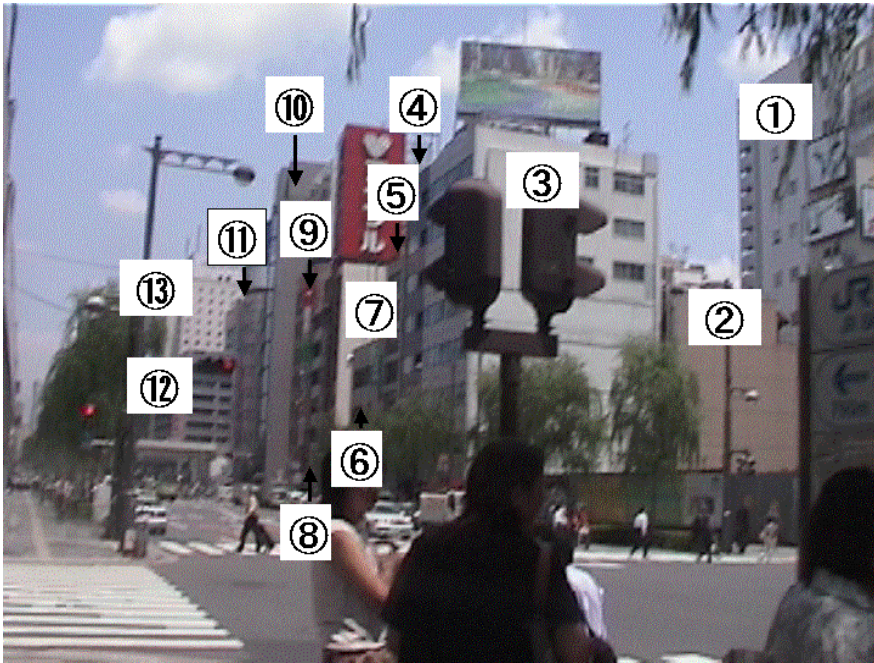


Fig. 7. Video Frame Captured at Time τ .

5 Conclusions and Future Works

In this paper, we proposed a new approach to indexing building objects captured by a video camera. The extraction of those buildings was performed using GPS and Gyro sensor data and a 3-dimensional city map data. It was shown that our algorithm works as intended. Future works include an implementation of the proposed indexing of video data, development of an effective storage scheme of video data, treatment of obstacles such as roadside trees and advertising displays (which make buildings invisible), introduction of a relative importance measure of building objects, and treatment of landmarks such as Tokyo Tower or Mt. Fuji.

Acknowledgements

The authors are thankful to Ms. Rei Ishiguro, who made significant contributions to this research during her stay at the Graduate School of Ochanomizu University. This

research was partly supported by a Grant-in-Aid for Scientific Research of MEXT in the Category of Exploratory Research (Grant number 14658089) on “Feasibility Study of Wearable Database Systems” (2002-2004) and the CREST (Core Research for Evolutional Science and Technology) Program of JST. DiaMap, the 3-dimensional city map, was provided by the Mitsubishi Corporation.

References

1. Gaughan, G., Smeaton, A., Gurrin, C., Lee, H., McDonald, K.: Design, Implementation and Testing of an Interactive Video Retrieval System, Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp.23-30 (November 2003)
2. Dunckley, L.: Multimedia Databases – An Object-Relational Approach –, 452p. (book), Addison-Wesley (2003)
3. Zhang, H., Low, C., Smoliar, S., Wu, J.: Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution, Proceedings of ACM Multimedia ‘95, pp.15-24 (November 1995)
4. Hampapur, A.: Semantic Video Indexing: Approach and Issues, ACM SIGMOD Record, Vol. 28, No. 1, pp. 32-39 (March 1999)
5. Zhou, W., Vellailal, A., Jay Kuo, C.-C.,: Rule-based Video Classification System for Basketball Video Indexing, ACM Multimedia Workshop, Marina Del Rey, California, USA (2000)
6. Ueda, T., Amagasa, T., Yoshikawa, M., Uemura, S.: Indexing Method of Image Data using Location and Time Information, Proceedings of the 12th Data Engineering Workshop (DEWS 2001), 125-86 (March 2001) (in Japanese)
7. Hori, T., Aizawa, K.: Context-based Video Retrieval System for the Life-Log Applications, Proceedings of the MIR’03, November 7, 2003, Berkeley, California, USA (November 2003)

VRML-Based 3D Collaborative GIS: A Design Perspective

Z. Chang and Songnian Li

Geomatics Engineering Program Dept of Civil Engineering, Ryerson University
350 Victoria Street, Toronto, Ontario, Canada M5B 2K3
{czheng,snli}@ryerson.ca

Abstract. Recent advance of technology has made it possible to view and simply manipulate 3D representations of spatial information over the Internet. 3D visualization, however, imposes many unique requirements to software design and implementation, as well as supporting data. Furthermore, current development of 3D modelling and visualization of spatial information mostly focus on single user platform. This paper presents our discussion on the design and development of an Internet-based 3D collaborative GIS system, which support real-time multi-user collaboration. While the prototype development is ongoing, some design issues are discussed based on our preliminary work. . . .

1 Introduction

3D spatial data models are the tools used to describe and represent 3D real world in computers. Traditional 3D models in geographical information systems (GIS) can be classified into two categories: digital elevation models (DEM) and triangulated irregular network (TIN) models [1]. Both are very suitable to represent the earth surface. However, these models have limitations in representing complex urban scenes used in urban planning and design, etc. (not a good city model). They are mostly not an object-oriented data model; or not suitable to be accessed on the Internet.

Virtual Reality Modelling Language (VRML), among other 3D Web technologies like MPEG-4 and Java3D and widely known as a modelling language for constructing web-based 3D models, has played a very important role in many fields such as computer games, scientific and geographic visualization. VRML uses plug-ins such as Cosmo Worlds or Cortona, allowing the virtual world to be downloaded and viewed from anywhere using a web browser [2]. Started as a specification to open the road towards platform independent 3D graphics on the Internet, VRML has now evolved to allow objects to be modified, rotated, and moved by means of interpolator nodes in VRML-based 3D worlds. More complex actions and interactivities are supported by script nodes which inline pieces of code, e.g., Java class and JavaScript. GeoVRML, extending VRML, enables representation and visualization of geographic data.

However, current VRML standards are limited to modelling scenes for “single-user experience” in the sense that simultaneous sharing of the same VRML model

by multi-users is not well addressed, not to mention the lack of group interactive capabilities of information exploration at the 3D object level and scenario evaluation (e.g., collaborative scenario construction in 3D scene). Extending VRML specifications to support group functions for “multi-user experience” seems to be one possible solution. However, this requires cultivated research and time to reach an agreement on the new specifications. A more widely adopted approach is to design and develop methods, processes, and supporting groupware systems that allow sharing of data among multiple participants and synchronization of operations on 3D worlds.

Research in collaborative 3D visualization and modelling has largely been in the domain of computer science, mostly under the umbrella of “collaborative virtual environments (CVE)”. This paper presents a brief review of these developments, especially some initial research efforts in the area of collaborative geographic visualization, followed by our emphasis on the requirements and design of synchronous 3D collaborative GIS systems, i.e., real-time sharing and manipulation of 3D models to solve problems.

2 Development Review

A number of research laboratories and commercial companies, such as Blaxxun and ParallelGraphics, are producing networked games and shared worlds which utilize central servers to share data among multiple participants. This alternative approach allows 3D worlds to be reliably scaled to accommodate several hundred participants.

Collaborative virtual environment [3] comprises applications such as multiplayer games and distributed battlefield simulations. They provide 3D spaces in which users can select and move a kind of puppet that serves as their representations, observes other users', and talks to others nearby through textual chat facilities, as illustrated in Figure 1. Other multiplayer games, such as Quake, have both human and non-human participants. Similar to single-user games and related research prototypes, multiplayer games serve as a domain of inspiration and validation for research into groupware. Some examples are the development of test case applications for groupware toolkits, such as Tic-Tac-Toe and CardTable on top of Rendezvous [4] and Tic-Tac-Toe, Solitaire, and Tetrominos on top of GroupKit [5].

These commercial systems, however, only focus on virtual worlds or virtual communities which are different from the real world. The real world with 3D geo-referenced data has special features associated with various map projections and coordinate systems, huge amount of data, different data resolutions and accuracies, and so on, for which VRML-based 3D GIS systems have to consider. Therefore, GeoVRML, an official Working Group of the Web3D Consortium, was formed with the goal of developing tools and recommended practice for the representation of geographical data using VRML. GeoVRML 1.1 specification has been defined to provide a number of extensions to VRML for supporting geographic applications.



Fig. 1. Multiplayer virtual worlds in worlds chat.

Several reported research projects focused on interactive 3D visualization of geographical data (e.g., [6] and [7]), although their results show the controlling of 3D GIS scenes and interactive querying on 3D objects only on a single-user basis. In terms of collaborative visualization, research effort in spatial community has so far focused on collaborative knowledge exploration and spatial decision support. More recently, the GeoVirtual Environments developed by the GeoVISTA Centre at Penn State University represents a significant effort towards 3D visualizations enabling the potential of same-time-different-place collaboration among scientists at remote locations as they explore complex spatial-temporal data [8]. The work is a part of ongoing effort in developing a framework to structure a systematic research in understanding an array of technology and human issues involved in major aspects of facilitating geo-collaboration.

3 Functional Requirements

Collaborative virtual worlds can be accessed by working teams who work on one project probably in different locations at the same time. The requirements of collaborative 3D depend on factors such as task, group, duration and context. Cooperative work may have many forms, e.g., multiplayer games, distributed battlefield simulations, net meeting applications and distance learning applications. These forms often change over time, even within the confines of a single project. To support such dynamic contexts, services typically should be comprehensive and flexible. For example, requirements may range between supports for: (1) same-time and different-time cooperative work; (2) same-place and different-place cooperative work; (3) single, discrete media such as text or graphics, and multiple media including continuous media such as audio and video; and (4) much freedom in the actions of users (permissive support) and coordination of the actions of users (restrictive support) [9].

On the other hand, collaborative 3D developers face many complexities in the collaborative development process. Compared to the development of single-user applications, the development of collaborative systems involves many additional technical issues from the area of distributed systems development, such as replication, consistency, concurrency, and communication protocols [9].

Some of the major functional and non-functional requirements for VRML-Based 3D Collaborative GIS based on the complexities mentioned above include: (1) Supporting both same-time, different-place cooperative work and single user work; (2) Internet accessible; (3) Collaborative viewing, controlling and object selection, and interactive modelling; (4) Integration of collaboration and communication tools, e.g. annotation, mark-up, audio/video conferencing, whiteboarding, etc.; (5) Integration of decision making and negotiation tools; (6) Easy to deployment; (7) Simple Floor Control management and session management

4 Architecture Design

4.1 Background

For applications that are able to be scaled to support many simultaneous users, peer-to-peer interactions are necessary on top of client-server query-response. As an example, the IEEE 1278.1 Distributed Interactive Simulation (DIS) protocol is a well-specified way to pass entity behaviour such as position, orientation, and collision, fire/detonate and other message types. Use of multicast networking enables scalable many-to-many communications, avoiding server bottlenecks. DIS is particularly effective at communicating physics information at interactive rates in real time [10]. The foundation of DIS data structure is a standard set of messages and rules, called Protocol Data Units (PDUs). An example of one of these data units is the Entity State PDU (ESPDU) which contains data about the position and velocity of an entity. The ESPDU also makes the type, position, orientation and appearance of an entity available to all other players of the distributed simulation. Figure 2 shows how networked DIS ESPDUs can be processed by a Java applet and passed to a VRML 2.0 scene. External authoring interface (EAI), a set of language-independent bindings, allows a VRML world to be accessed and manipulated from an outside environment. EAI defines an interface between a VRML world and an external environment. It contains a set of functions of the VRML browser that the external program can call to affect or get parameters from the VRML world. This contribution deals only with interface between a Java applet on a HTML page and a VRML world opened in a viewer embedded in the same page (Figure 3).

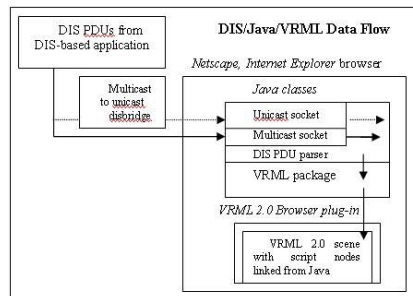


Fig. 2. Example data flow from DIS ESPDUs (source: [11]).

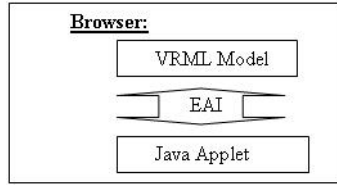


Fig. 3. External authoring interfaces.

4.2 Related Work

There are several other Java-based collaboration systems that have been seen over the last few years. Examples include the NCSA Habanero, the Java Collaboration Environment (JCE), and Java Applets Made Multi-user (JAMM).

Habanero[12] is a collaborative framework and environment containing a set of applications. The Habanero client, server and applications provide the necessary environment to create collaborative workspaces and virtual communities. The server hosts sessions and connects the clients that interact with the sessions using a variety of applications called Hamlets. Sessions can be recorded, persistent, access restricted and even anonymous. The client provides the interface to define, list, create, join and interact with a session. Unfortunately, the Habanero project has been discontinued.

Java Collaborative Environment[13] is a framework for shared interactive multimedia applications in heterogeneous systems. A collaborative mechanism to intercept, distribute and recreate the user events has been developed to allow Java applications to be shared transparently. The approach is based on the replicated tool architecture.

JAMM [14] is an application-sharing system that allows multiple users to simultaneously work in a legacy, single-user application. JAMM is an alternative implementation based on an object-oriented replicated architecture where certain single-user interface objects are dynamically replaced by multi-user extensions.

4.3 System Design Based on Replicated Architecture

There are three kinds of architectures in collaborative applications: centralized architecture, replicated architecture, and distributed architecture [15]. Since the replicated architecture is adopted in our current research project, most discussions hereafter are related to this architecture.

A centralized architecture provides only one application, and distributes copies of the GUI (view and controller) by sending window system events to all participating client machines. In a replicated architecture the entire application is installed and run (i.e., replicated) on each client machine; and some means of synchronization between them is provided. A distributed architecture is characterized by the distribution of the Model-View-Control (MVC) components across multiple hosts. Typically, the model lives on a shared server and each client has its own view and controller.

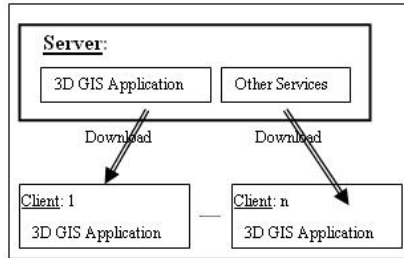


Fig. 4. Replicated architecture.

Figure 4 illustrates a replicated architecture used in a collaborative system. This system is also a client-server based networking system that synchronizes movement and events within virtual environments over an IP network. When the user accesses the 3D GIS application website with a web browser (e. g., IE Explore or Netscape), the 3D GIS Application in the form of Java component is downloaded into the local machine and run in the browser. So every browser has the same MVC components. The 3D collaborative GIS application consists of four parts, as shown in Figure 5. The first part, named VRML component, handles VRML model and the second part renders the content of VRML world through a 3D component, while the “2D-Viewer component” part shows 2D content based on VRML world. The fourth part, called “Collaborative component” synchronizes the event messages sent by the client and received by the server. There is also a Collaborative Service component on the server tier which handles the synchronization of operations among clients.

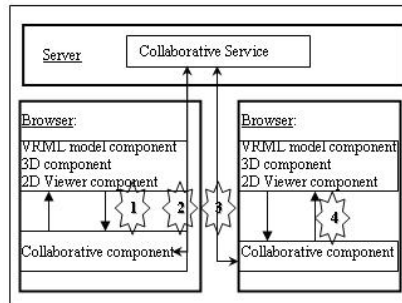


Fig. 5. Data flow.

4.4 Process of Event Sending

Based on the above architecture, there are two types of events triggered by the users. One type is no-3D events like menu clicks, 2D events, etc. which are sent and reconstructed and handled through Java components. Another type is 3D events which are handled by EAI-based java component first and sent through DIS protocol.

The process of event sending based on the replicated architecture is shown in Figure 5 above and is briefly described as follows.

When a user operates GUIs of Java applets that handle 3D or 2D contents in the client named host, the events are first triggered and at the same time sent to the Collaborative component at the same client. These events then are sent to Collaborative Service in the server side. The Collaborative Server distributes the events to the Collaborative component of every client who joins the same session. After reconstructed, these events are finally sent to 3D or 2D Plug-in to realize the same functions as host client does.

4.5 Framework of Collaborative 3D GIS Systems

The 3D collaborative GIS system follows a client-server network model and a replicated architecture (see Figure 6). While the client is simply a web browser with downloadable Java applets, the server tier consists of middleware and database layer.

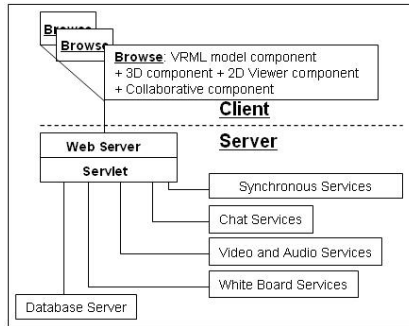


Fig. 6. Framework of multi-user collaborative 3D system.

Middleware running on the server side ships out VRML and HTML content according to the types of services required. Chat services, whiteboard services, video and audio services are coordinated through the Synchronous Services component. Database Server is used as storage for persistent structured information. This information includes user records, world information, work environment and GIS 2D and 3D data etc.

5 Autonomous Systems

The designed replicated architecture has been tested on a simple client-server Java application. The server (see Figure 7) based on Java Shared Data Toolkit (JSDT) which is a java API from Sun Microsystems, Inc. is used to receive the messages from clients and sends them to the identified clients. Through joining in a server, a client is able to connect to other clients. One server may have several sessions and one session also may consume several channels.

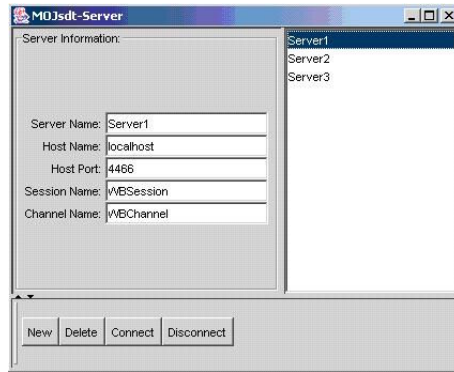


Fig. 7. Server of multi-user collaborative 3D prototype application.

The client application (see Figure 8) includes four components: GIS component based on MapObjects Java Edition, Java3D, JSDT and VRML Loader. Java3D handles 3D content; MapObjects Java handles GIS data; VRML loader loads VRML model; JSDT sends messages among clients.

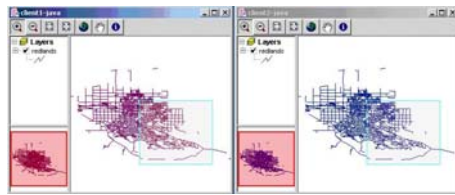


Fig. 8. Clients of multi-user collaborative 3D prototype application.

When a server already exists, a client is able to find the server through sending a message including a Server name and an IP address. After receiving the message, the server sends the related information which includes Host Name, Port Number, Session Name and Channel Name, etc. to the client. Finally the connection is created between the two parts.

Two more tasks need to be done next. First, assign Floor Control right to the client. If the client is the first to enter the session, the right of Floor Control should be assigned to the client. Floor Control helps to coordinate joint and competing activities among people and their interacting computational processes, such as regulating turn-taking in conversations or write-updates on shared files, preserving coherency of local and remote information[16]. This client also is able to give the right to identified client in this session. But only one client has the right. Second, handle later-coming clients. If the client is not the first to join in the session, the information (e.g. Map extent, layers information, etc.) from the other clients who are in the same session can be downloaded in the client. Therefore all the clients in the session are kept synchronized.

The problems about system reliability, scalability need to be further worked next.

6 Summary

VRML models can be well adapted into collaborative 3D GIS systems because of their special features. These features help VRML for more complex interactive jobs. To support collaborative VRML model viewing and manipulating, a multi-tier client-server structure is required for a collaborative 3D GIS system, which uses designed the replicated architecture in our study. The most complicated tier is Synchronous Services in the middleware which receives and broadcasts real-time interaction messages.

References

1. Abdelguerfi, M., Wynne, C., Cooper, E.: Representation of 3-D Elevation in terrain databases using hierarchical triangulated irregular Network: a Comparative Analysis, *Geographical Information Science*, December, Vol. 12, No. 8. (1998) 853-873
2. Al-Kodmany, K.: Visualization Tools and Methods in Community Planning: From Freehand Sketches to Virtual Reality, *Journal of Planning Literature*. Vol. 17. No. 2. (2002)
3. Greenhalgh, C.: Creating large-scale collaborative virtual environments. EC-SCW'97 OOGP workshop, <https://doc.telin.nl/dscgi/ds.py/Get/File-7704/greenhalgh.pdf>
4. Hill, R.D., Brinck, T., Rohall, S.L., Patterson, J.F., Wilner, W.: the rendezvous architecture and language for constructing multiuser applications, *ACM Transactions on Computer-Human Interaction*, 1 (June 1994), 2, p. 81-125,
5. Roseman, M., Greenberg, S.: GroupKit: A groupware toolkit for building real-time conferencing applications. In [CSCW92], (1992) 43-50,
6. Campbell, B., Colline, P., Hadaway, H., Hedley, N., Stoermer, M.: Web3D in Ocean Science Learning Environments: Virtual Big Beef Creek. *Proceedings of Web3D'02*. February 24-28, (2002), Tempe, Arizona, USA
7. Huang, B., Jiang, B., Lin, H.: An integration of GIS, virtual reality and the Internet for visualization, analysis and exploration of spatial data. *International Journal of Geographic Information Science*. Vol. 15. No. 5. (2001) 439 - 4
8. MacEachren, A. M., Brewer, I., Steiner, E.: GEOVISUALIZATION TO MEDIATE COLLABORATIVE WORK: Tools to Support Different-Place Knowledge Construction and Decisionmaking. *Proc. of the 20th Int. Cartographic Conf.*, Beijing, China, August (2001) 6-10.
9. Hofte, G.: Working Apart Together - Foundations for Component Groupware. Enschede, The Netherlands. (1998)
10. Brutzman, D.: The Virtual Reality Modeling Language and Java. *Communications of the ACM*, 41(6), (1998) 57-64
11. Brutzman, D., Zyda, M., Watsen, K., Macedonia, M.: virtual reality transfer protocol (vrtp) Design Rationale, Presented at Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises (WET ICE): Sharing a Distributed Virtual Reality, assachusetts Institute of Technology, Cambridge Massachusetts, June 18-20 1997.
12. Chabert, A., Grossman, E., Jackson, L., Pietrowicz, S., Seguin, C.: Java Object-Sharing in Habanero. *Communications of the ACM*. Vol. 41. No. 6. (1998) 69-76
13. JCE, 2004. Java Collaboration Environment website, <http://snad.ncsl.nist.gov/madvtg/Java/overview.html> (accessed on May 20, 2004)

14. JAMM, 2004. Flexible Java Applets Made Multiuser website, <http://simon.cs.vt.edu/JAMM/> (accessed on May 20, 2004)
15. Suthers, D.: Architectures for Computer Supported Collaborative Learning, IEEE International Conference on Advanced Learning Technologies (ICALT 2001). (2001)
16. Dommel, H., Garcia-luna-Aceves, J.J. : Floor Control for Multimedia Conferencing and Collaboration, Multimedia Systems, vol. 5, no. 1, pp. 23–38, 1997.

Arrival Time Dependent Shortest Path by On-Road Routing in Mobile Ad-Hoc Network

Kyoung-Sook Kim, So-Young Hwang, and Ki-Joune Li

Department of Computer Science and Engineering
Pusan National University, Pusan 609-735, South Korea
{ksookim,youngox,lik}@pnu.edu

Abstract. Arrival time dependent shortest path finding is an important function in the field of traffic information systems or telematics. However large number of mobile objects on the road network results in a scalability problem for frequently updating and handling their real-time location. In this paper, we propose a query processing method in MANET(Mobile Ad-hoc Network) environment to find an arrival time dependent shortest path with a consideration of both traffic and location in real time. Since our method does not need a centralized server, time dependent shortest path query is processed by in-network way. In order to reduce the number of messages to forward and nodes to relay, we introduce an *on-road routing*, where messages are forwarded to neighboring nodes on the same or adjacent road segments. This routing method allows to collect traffic information in real time and to reduce the number of routing messages. Experiments show that the number of forwarded messages is reduced in an order of magnitude with our *on-road routing* method compared to LAR-like method. At best, our method reduces about 57 times less messages.

1 Introduction

MANET consists of mobile nodes that communicate with each other, in the absence of a fixed infrastructure. Mobile devices construct network spontaneously when they need to communicate in MANET. The technology related with MANET is composed of application software, routing, transport layer, medium access control and physical layer for various applications. Especially, many routing protocols [1–3, 5, 6] have been proposed for MANET, with the goal of achieving efficient routing because of unpredictable topology changes caused by mobility and limited wireless transmission range.

Telematics is one of important application area based on MANET and location-based services. Telematics services gather real-time traffic data from vehicles and provide information such as location, traffic, accident, emergency and the shortest path for drivers and passengers. In order to offer these services, we may need a fixed infrastructure such as loop sensors and centralized hosts to store and process data. In this paper, we propose a query processing mechanism, called *on-road routing*, with the consideration on real-time traffic information of large

number of vehicles. In particular, we focus on the method that process arrival time dependent shortest path query in MANET without central servers and traffic sensors such as loop-sensor on the road networks [10–12]. The main idea of our approach lies in a routing message that includes query predicates based on the road connectivity and on data gathering method in real time from vehicles on the road by ad hoc network. Therefore, we should reduce unnecessary flooding messages by pruning mobile nodes which are not on the same or neighbor road segments.

The rest of this paper is organized as follows. Section 2 discusses motivation and related research in the area. In section 3, we describe proposed approach for routing messages in MANET based on the road networks. Section 4 includes the performance evaluation of the proposed method. Finally, we conclude this paper in section 5.

2 Motivation and Related Work

Many researchers have been investigated the shortest path algorithm based on the road networks. Most of them concentrate on computational time complexity or memory complexity to improve the algorithm [8, 9]. To find out optimal route using this algorithm, real-time traffic information must be stored in the server [10–12]. But it is hard to gather information of mobile nodes without the fixed infrastructure and the performance is degraded to process only in centralized hosts. Therefore we propose a routing mechanism for query process in MANET. Routing protocols in ad hoc networks are categorized into routing with location information and without roughly. Protocols without location information are DSR [1], AODV [2], ZRP [3] and so on. Then we focus on routing protocols that take location information into account.

Recently researches on routing and applications in mobile ad hoc network are increasing enormously based on location information [4–7, 13–16]. First of all, Location Aided Routing (LAR) is a representative example that utilizes location information to improve the performance of routing protocols for ad hoc networks. By using location information, the LAR limits the search for a route to the so-called request zone, determined based on the expected location of the destination node at the time of the route discovery. Simulation results indicate that using location information results in significantly lower routing overhead; as compared to an algorithm without location information [5].

The next protocols are that routing dependent on applications such as information dissemination in multihop inter-vehicle networks [13] or multicast protocol in ad hoc networks – inter-vehicle geocast [14, 15]. The former is motivated by the use of ad hoc networking technologies for vehicle information exchange. If the geographic positions of the network nodes are known, better performance can be achieved with the utilization of so called geocast routing algorithms. Authors proposed a context-based geocast routing protocol that is based on AODV (Ad hoc On Demand Distance Vector protocol). It uses other attributes, such as the speed and direction of the nodes besides location for forwarding messages.

The set of nodes that are interested in the context of a message does not depend only on the position, but also driving direction, destination and even user profile, vehicle type etc. It mainly depends on the situation and the content of the message which attributes are relevant [13]. The latter – Inter-Vehicle Geocast (IVG) – is mainly designed for effective alarm message dissemination in the ad hoc network of vehicles in a highway. IVG is based on geographic multicast, which consists in determining the multicast group according to the driving direction and the positioning of vehicles. The multicast is restrained to the so-called risk areas. First, a broken vehicle (or accident) begins to broadcast an alarm message to inform the other vehicles of its situation. Since the accident vehicle can just inform its one-hop neighbors, some other vehicles have to rebroadcast the alarm message to inform the vehicles located at more than one hop from the accident. The vehicle that performs the rebroadcast is called relay. Relays in IVG are designated in fully distributed manner. The way with which a node is designated as relay is based on distance defer time algorithm. The node that receives an alarm message does not rebroadcast it immediately but has to wait some time to make a decision about rebroadcast. When the deferred time expires, if it does not receive the same alarm message from another node behind it, it deduces that there is no relay node behind it. Thus it has to designate it self as a relay and starts to broadcast the alarm messages in order to inform the vehicles which could be behind it. The process of message dissemination with IVG depends on the rate of vehicles equipping GPS (Global Positioning System) in the road. It is believed that the performance of IVG is dependent on the portion of vehicles which are not equipped with GPS [14].

These researches are analogous with ours in improving routing performance using location information and in role-based multicasting based on the road or highway. However, LAR considers just routing in Euclidian space and IVG concentrates on dissemination of alarm messages such in case that car accident is occurred and so on. Therefore, we should take account of the space embedding a road network not Euclidian space to find the shortest path depending on estimated arrival time. We expect a performance improvement through restricting the message forwarding area to the same or adjacent road segments. For what is called *on-road routing* considering query predicates. It can not only disseminate alarm messages but also process other queries besides the shortest path through tuning query condition in the message. The mechanism is specified following routing on the road networks in detail.

3 On-Road Routing

In the following we present our proposed routing algorithm for query processing in mobile ad hoc network based on the road networks. The main issues to consider include (1) how to define ad hoc network domain and (2) routing strategy to improve performance in such road networks.

3.1 Basic Concept

Road networks consist of links which are the polyline set bounded by two crosses. Figure 1 shows a mobile node which is included in an ad hoc network and travels on the road.

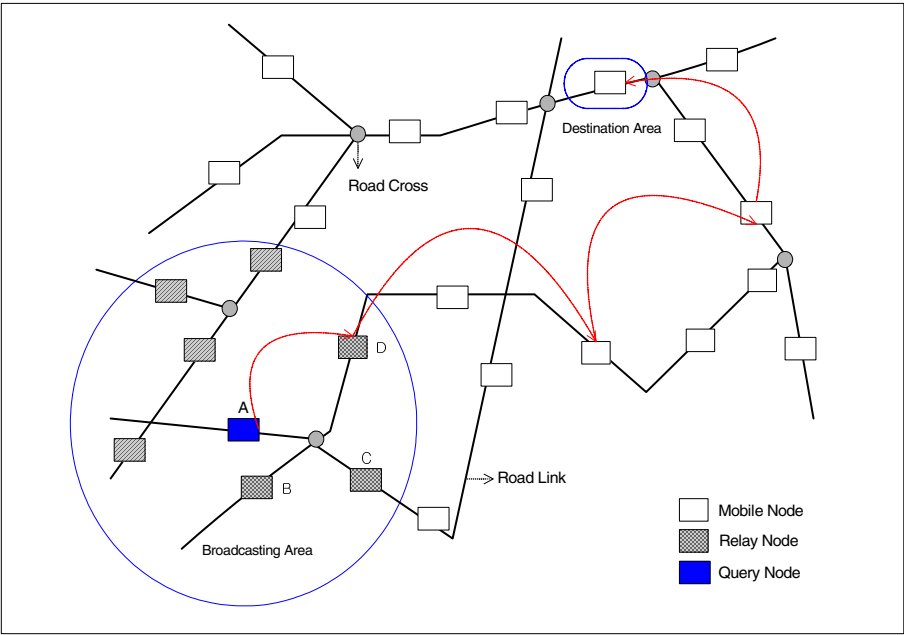


Fig. 1. Mobile nodes on the road networks.

Query processing is performed as follows. Suppose that a query is given as “find arrival time dependent shortest path from a source to a destination area” in node A as shown by figure 1. Query node A generates a route request message including query condition and broadcasts the message. In figure 1, seven nodes receive the message since they are included within the broadcast domain of query node A and decide whether to rebroadcast the message or not. According to legacy routing algorithm that operates under Euclidean space all the seven nodes rebroadcast the routing request message. But in our mechanism, only the nodes which travel on the same or adjacent road of query node A relay the message. Therefore, three mobile nodes, which are node B, C and D, take a part in forwarding the route request message. This procedure is performed recursively until it finds the destination. The destination node finally replies in reverse order through a path traversed by the routing procedure.

There are assumptions to process query like this. First, nodes are quasi-static during the short period of a route finding phase. Second, there is at least one node on the same or adjacent road of a query node (or a relay node) in

the broadcast domain. Each node does not manage it's neighbor nodes. Two nodes are regarded as neighbors if they can communicate with each other over a wireless link. The broadcast domain is defined by the transmission range of wireless media. For example, IEEE 802.11b wireless lan has a finite range for transmission – around 100 meters indoors and 300 meters outdoors. This range can be extended through transmission power control, antenna performance and so on. Finally, each node maintains road information which belongs to the road traveled by the node. For the last one, we introduce a labeler. Labelers are located in each road link. It has geometry data of road link and adjacent labeler list. Labeler offers road information when a mobile node enters the road. As a result mobile nodes can maintain road information and use it in routing.

3.2 Data Structures

There are two objects. One is the labeler and the other is the mobile node for our routing algorithm. Function of labeler is as follows. A labeler maintains its road information and its adjacent labeler list. When a new mobile node enters a road segment, the labeler which is included in the road segment hands over its data to the mobile node. A mobile node can manage its position data through maintaining labeler's information and offset of distance from the starting position of a road. Figure 2 represents data structure of a labeler.

Labeler ID		
Road Information	Geometry	Coordinates
		Distance
	Topology	Start Cross
		End Cross
Adjacent Labeler ID List		

Fig. 2. Data structure of a labeler.

Mobile Node ID
Offset (Position)
Speed
Direction
Labeler Information

Fig. 3. Data structure of a mobile node.

Mobile node can maintain its position data through labeler information and distance offset as mentioned previously. When a mobile node enters a new road segment, it replaces road information from a labeler which is included in the road segment. Figure 3 depicts data structure of a mobile node. It can contain additional information or user profile dependent on applications.

3.3 On-Road Routing Strategy

Basic approach of our routing mechanism is almost the same with route discovery using flooding. But we adapt location information, road information and query condition to reduce the number of nodes to whom route request is propagated since our mobile ad hoc network is based on the road networks. This is a kind of the flooding and pruning algorithm based on the road networks and query condition. In the following we describe the detailed steps of proposed routing strategy (Figure 4 shows our *on-road routing* algorithm).

Step 1: A source node needs to find a route to a destination node or destination area, defines request zone to restrict message flooding. Then it generates a route request message. The message contains request zone information and query condition which contains request of user. After that source node broadcasts a route request message to all its neighbors – two nodes are regarded as neighbors if they can communicate with each other over a wireless link.

Step 2: A node, say relay node, on receiving a route request message, compares the sequence number with prior received messages' to detect repeated reception of a route request. Then the relay node compares request zone with its own position information if it is included in the request zone or not. And then it compares road information with its own if it is along the road or not. After that it processes query if it is needed during routing. This is a selection factor to avoid redundant transmission of route requests when the relay node receives multiple request messages with the same sequence number simultaneously in an allowed time interval. An example is presented in the following simulation. To find a arrival time dependent shortest path, the travelling time must be calculated during the routing. The relay node selects the best result after calculating. Finally it compares the desired destination with its own information such as position or road information.

Step 3: If there is a match, it means that the request is for a route to itself. Otherwise, it decreases TTL(Time To Live) value. If the TTL value is not equal to zero it rebroadcasts the route request message to its neighbors after appending its address to payload which includes the path traversed by the request.

Step 4: Repeat Step 2 until it finds a match.

```

On-Road Routing Algorithm
Input
  m: received message
  N: information of current node

Begin
  If check m.sequence_number
    /* detect multiple receptions of the same route request */
    discard the message; /* sequence number pruning */

  If position of current node not contained in request zone
    discard the message; /* request zone pruning */

  If current node not on same or adjacent road of previous node's
    discard the message; /* road information pruning */

  ProcessQuery; /* process query predefined by user */

  If position of current node is destination
    SendReply; /* find a match */

  Else decrease TTL; /* prevent infinite flooding */
    If TTL equal to zero
      return;
    Else
      add address of current node to payload;
      /* a path traversed by the request */

      BroadcastMessage;
      /* rebroadcast route request message */
End

```

Fig. 4. *On-Road Routing* algorithm.

It is possible that the destination will not receive a route request message (for instance, when it is unreachable from the source or route requests are lost due to transmission errors). In such cases, the source needs to be able to reinitiate route discovery. Therefore, when a source initiates route discovery, it sets a timeout. During the timeout interval, if a route reply is not received then a new route discovery is initiated. The route request messages for this route discovery will use a different sequence number than the previous route discovery. Figure 5 illustrates message format for routing request and query processing.

4 Experiments

In order to evaluate the performance of the proposed method, we established a model of road networks and mobile nodes which travel along the road. There

Message Type	Sequence Number	TTL
Address (broadcast address or popped up address from palyload)		
Request Zone		
Destination Information (Query Condition)		
Source Information		
Intermediate or Final Result of Query Condition		
Payload (a path traversed by the request)		

Fig. 5. Message format for *on-road routing*.

are measurement factors for routing in mobile ad hoc network such as energy efficiency, the number of messages flooded, bandwidth consumption, latency and so on. In this simulation, we concentrate the number of nodes to whom route request is propagated according to each pruning strategy.

4.1 Simulation Scenario and Framework

Figure 6 illustrates the road networks in a region of Seoul in Korea. We generated a synthetic data set consisting of approximately 10,000 moving objects as mobile nodes using the benchmark tool of Brinkhoff on the real road networks [17]. We regard generated moving objects as a series of static objects which are obtained at arbitrary time slice. We assume that the velocity of mobile nodes is identical if they are on the same road segment.

The radius of each broadcast range is set with 0.02%, 0.03% and 0.04% of the road networks' Euclidean width and TTL is fixed with 30 heuristically. Query conditions are estimating shortest arrival time from a source node to a destination node (or area) and finding the path with limited road information – which is described in section 3.1. As mentioned in the prior routing strategy, arrival time is calculated during the routing. The route request message is delivered through the path which has shorter arrival time. Once the message arrives at the destination, the shortest path is found. If the destination receives multiple messages, it selects the best one by calculating arrival time. Estimated arrival time and the routed path are included in the routing message. The destination node replies in reverse order through the path traversed by the request. In our simulations, we do not model the delays that may be induced by multiple nodes which attempt to transmit data simultaneously. Transmission errors and congestion are also not considered.

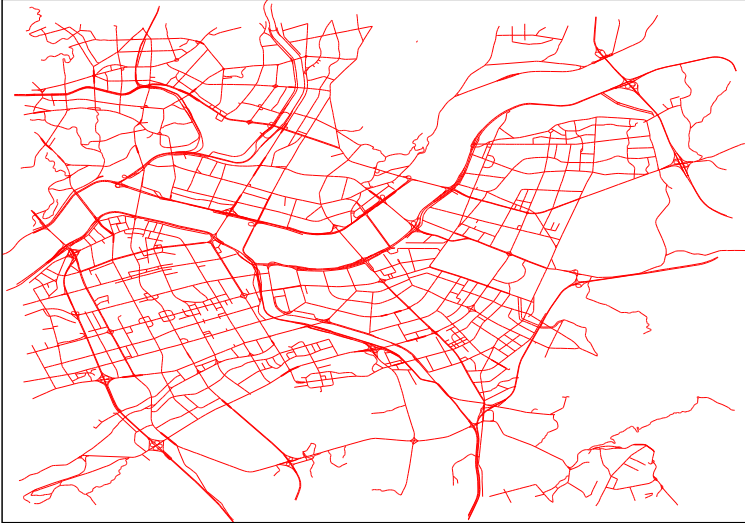


Fig. 6. Road networks in a region of Seoul in Korea.

4.2 Experiment Results

In this simulation, we concentrate the number of nodes to whom route request is propagated according to each pruning strategy. We use location information, road connectivity and query condition to reduce the number of messages. Figure 7 shows average number of messages discarded by each pruning strategy. Road information is dominating factor to reduce the number of messages in our simulation. This result is significant since routing along the road segments is reasonable in road networks.

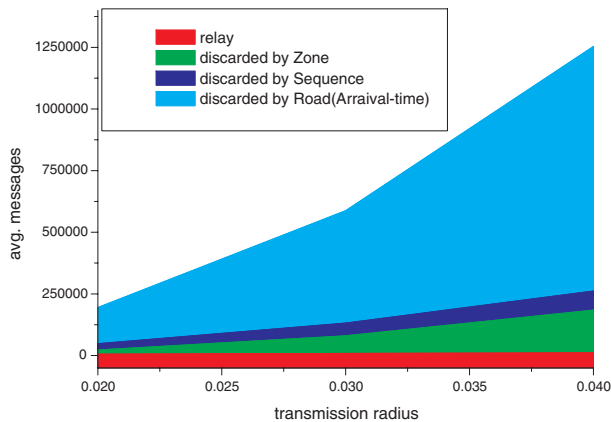


Fig. 7. Average number of messages pruned by each strategy.

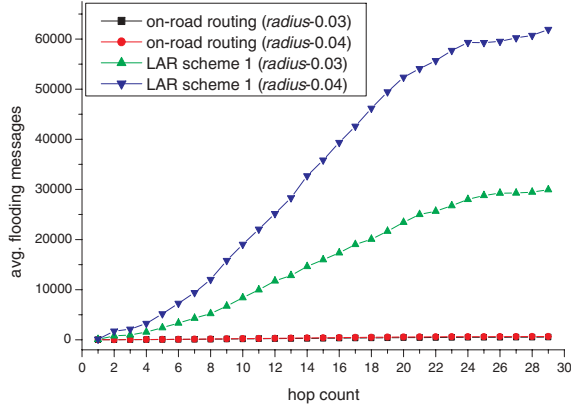


Fig. 8. Performance comparison on-road routing with LAR scheme 1 by request zone.

Figure 8 depicts the performance comparison of our approach with LAR scheme 1 – which is flooding with request zone. Differently from LAR scheme 1, our method has been less affected by the radius of broadcast area. Because our method restricts the message forwarding area to the same or adjacent road segments. These results signify that our proposed approach can help energy efficiency, bandwidth utilization and etc. since communication dominates a node’s energy consumption and collision affects bandwidth utilization.

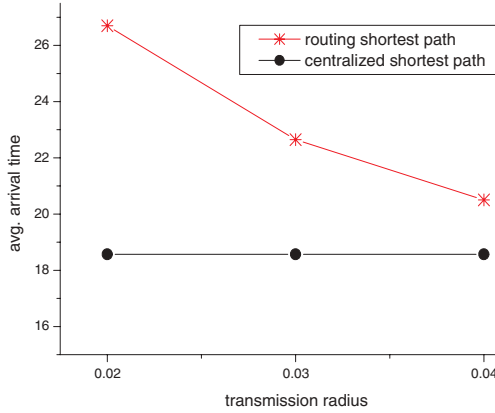


Fig. 9. Estimated arrival time by transmission range.

Our results make a little difference in the optimal shortest path due to the limitation of the search space which is defined by broadcast domain [Figure 9]. If we use centralized server which manages all mobile nodes, we can obtain optimal result. But large number of mobile objects on the road network results in a scalability problem for updating and handling location data in real time. Moreover it is not suitable in this kind of ad hoc environment without infrastructure.

5 Concluding Remarks

In this paper, we considered query processing methods of moving objects on the road in MANET, and proposed a routing method, call *on-road routing*. Unlike other routing methods, such as LAR [5], and GPSR [6], our method restricts the message forwarding area to the same or adjacent road segments for two reasons. First, we are interested in only traffic and location information of vehicles on these roads. Second, by pruning the messages not travelling these roads, we can significantly improve the routing performance during query processing.

In particular, we focus on the query processing for arrival time dependent shortest path in mobile ad hoc network. An in-networking query processing method has been proposed for this type of query. And we performed experiments to verify our conjecture that the performance of our method is superior to other routing methods proposed based on geographic information in Euclidian space. The results show that the number of forwarding messages is significantly reduced by applying *on-road routing* method compared to methods based on LAR scheme-1. Our method reduces the number of flooding messages in an order of magnitude. At best, it reduces about 57 times less messages.

Although we have proposed an on-road routing method, we believe that it can be improved by considering several factors, such as dynamic TTL, vehicle velocity, and constraints of the road network. Future work therefore includes improvement of our *on-road routing*.

Acknowledgment

This research was partially supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce, Industry and Energy of the Korean Government, by IRC(Internet Information Retrieval Research Center) in Hankuk Aviation University. IRC is a Kyounggi-Province Regional Research Center designated by Korea Science and Engineering Foundation and Ministry of Science & Technology.

References

1. D. B. Johnson and D. A. Maltz, Dynamic Source Routing in Ad Hoc Networks, Mobile Computing, Kulwer Academic Publishers, Volume 353, pp.152-181, 1996
2. C. E. Perkins and E. M. Royer, Ad hoc On-Demand Distance Vector Routing, Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100, 1999
3. Z. J. Haas and M. R. Pearlman, The performance of query control schemes for the zone routing protocol, IEEE/ACM Transactions on Networking 9(4), pp.427-438, 2001
4. Y.-B. Ko and N. H. Vaidya, Using location information in wireless ad hoc networks, IEEE 49th Vehicular Technology Conference, Volume 3, pp.1952-1956, 1999
5. Y.-B. Ko and N. H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, Wireless Networks 6(4), pp.307-321, 2000

6. B. Karp and H. T. Kung, GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp.243–254, 2000
7. M. Mauve, J. Widmer, H. Hartenstein, A Survey on Position-Based Routing in Mobile Ad-Hoc Networks, IEEE Network, 15(6), pp.30–39, 2001
8. R. K. Ahuja, K. Mehlhorn, J. Orlin, R. E. Tarjan, Faster algorithms for the Shortest Path Problem, Journal of Association of Computing Machinery 37(2), pp.213–223, 1990
9. F. B. Zhan, Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures, Journal of Geographic Information and Decision Analysis 1(1), pp.69–82, 1997
10. O. Wolfson, L. Jiang, A. P. Sistla, S. Chamberlain, N. Rishe, M. Deng, Databases for Tracking Mobile Units in Real Time, Proceedings of the 7th International Conference on Database Theory, pp.169–186, 1999
11. G. Trajcevski, O. Wolfson, B. Xu, P. Nelson, Real-Time Traffic Updates in Moving Objects Databases, Proceedings of the 13th International Workshop on Database and Expert Systems Applications , pp.698–704, 2002
12. H.-D. Chon, D. Agrawal, A. E. Abbadi, FATES: Finding A Time dEpendent Shortest path, Proceedings of the 4th International Conference on Mobile Data Management, pp.165–180, 2003
13. T. Kosch, C. Schwingenschlogl, Li Ai, Information dissemination in multihop inter-vehicle networks, Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, pp.685–690, 2002
14. A. Bachir, A. Benslimane, A multicast protocol in ad hoc networks inter-vehicle geocast, Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, Volume 4, pp.2456–2460, 2003
15. A. Bachir, A. Benslimane, Towards supporting GPS-unequipped vehicles in inter-vehicle geocast, Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, pp.766–767, 2003
16. B. An, S. Papavassiliou, An architecture for supporting geomulticast services in mobile ad-hoc wireless networks, IEEE Military Communications Conference, Communications for Network-Centric Operations: Creating the Information Force, Volume 1, pp.301–305, 2001
17. T. Brinkhoff, A framework for generating network-based moving objects, GeoInformatica 6(2), pp.153–180, 2002

Author Index

- Arikawa, Masatoshi 38
- Bertolotto, Michela 181
- Boucelma, Omar 81
- Brisaboa, Nieves R. 94
- Chang, Z. 232
- Colonna, François-Marie 81
- Deng, Ke 151
- Doyle, Julie 181
- Gautam, Arvind 51
- Han, Qiang 181
- Hong, Bonghee 64
- Huang, Xuegang 120
- Hwang, Chong-Sun 26
- Hwang, So-Young 242
- Jensen, Christian S. 120
- Jin, Hong Tae 110
- Joo, Inhak 1
- Kang, Hye-Young 136
- Kim, Chang-Soo 51
- Kim, Do-Hyoung 206
- Kim, Dong Seong 110
- Kim, Donghyun 64
- Kim, Jong-Woo 51
- Kim, Kyoung-Sook 242
- Kim, Kyung-Chang 167
- Kim, Mijeong 1
- Kim, Minsoo 1
- Kwon, Yong-Jin 206
- Lee, Eunkyuu 1
- Lee, Yugyung 51
- Li, Ki-Joune 136, 242
- Li, Songnian 232
- Lim, Bog-Ja 136
- Luaces, Miguel R. 94
- Masunaga, Yoshifumi 221
- Noaki, Kouzou 38
- Paramá, José R. 94
- Park, Jong Sou 110
- Park, KwangJin 26
- Sato, Yukiko 221
- Song, MoonBae 26
- Tanaka, Katsumi 14
- Tezuka, Taro 14
- Viqueira, Jose R. 94
- Weakliam, Joe 181
- Wilson, David 181
- Yun, Suk-Woo 167
- Zhang, Qing-Nian 195
- Zhou, Xiaofang 151