Sara Irina Fabrikant
Tumasch Reichenbacher
Marc van Kreveld
Christoph Schlieder (Eds.)

LNCS 6292

Geographic Information Science

6th International Conference, GIScience 2010
Zurich, Switzerland, September 2010
Proceedings

Springer

Lecture Notes in Computer Science 6292

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

Sara Irina Fabrikant
Tumasch Reichenbacher Marc van Kreveld
Christoph Schlieder (Eds.)

Geographic Information Science

6th International Conference, GIScience 2010
Zurich, Switzerland, September 14-17, 2010
Proceedings

Springer

Volume Editors

Sara Irina Fabrikant
University of Zurich, 8057 Zurich, Switzerland
E-mail: sara.fabrikant@geo.uzh.ch

Tumasch Reichenbacher
University of Zurich, 8057 Zurich, Switzerland
E-mail: tumasch.reichenbacher@geo.uzh.ch

Marc van Kreveld
Utrecht University, 3508 TB Utrecht, The Netherlands
E-mail: marc@cs.uu.nl

Christoph Schlieder
University of Bamberg, 96047 Bamberg, Germany
E-mail: christoph.schlieder@uni-bamberg.de

Library of Congress Control Number: 2010932515

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

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-642-15299-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15299-3 Springer Berlin Heidelberg New York

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

Since its inception in Savannah, Georgia (USA) in 2000, the highly successful GIScience conference series (www.giscience.org) has regularly attracted over 250 researchers from all over the world whose common interest lies in advancing the research frontiers of fundamental aspects of the production, dissemination, and use of geographic information. The conference is bi-annual and brings together leading researchers from all cognate disciplines reflecting the interdisciplinary breadth of GIScience, including (but not limited to) geography, cognitive science, computer science, engineering, information science, mathematics, philosophy, psychology, social science, and (geo)statistics.

Following the, literally breathtaking, conference in Park City, Utah (USA) at 2103m, the sixth GIScience 2010 conference returned to Europe for the second time. The 2010 conference was held in Zurich, Switzerland, a place nominated repeatedly as the world's most livable (if not cheapest!) city. Zurich is also a GIScience landmark, as in 1990 one of the founders of the GIScience conference series, Dr. Michael Goodchild, delivered a memorable talk setting out how fundamental research on GISystems could turn into GIScience at the very same conference location during the Spatial Data Handling Symposium.

To accommodate the variety of publication cultures within an interdisciplinary research community, GIScience 2010 provided, as in previous years, two refereed submission tracks: full papers (15 pages), and extended abstracts (1500 words). In all, 87 full papers were each thoroughly reviewed by at least three Program Committee members, of which 22 were selected for a 30-minute presentation at the conference and included in this volume. Of the 146 submitted extended abstracts describing work in progress that were submitted and reviewed by Program Committee members, 98 were accepted for either oral or poster presentation. All extended abstracts appeared in a single proceedings volume disseminated as a USB stick at the conference, and are permanently archived online at the GIScience 2010 website (www.giscience2010.org).

The GIScience 2010 proceedings serve as impressive evidence of the maturity and continuing growth of this still young, but vibrant, and interdisciplinary research arena. Despite the economic downturn, 2010 saw the second highest total number of submissions (after Münster in 2006) in the series. The breadth of the topics in this volume also reflects the breadth of the disciplines involved in fundamental research related to geographic information. While traditional research topics such as spatio-temporal representations, spatial relations, interoperability, geographic databases, cartographic generalization, geographic visualization, navigation, spatial cognition, etc. are alive and well in GIScience, research on how to handle massive and rapidly growing databases of dynamic space–time phenomena at fine-grained resolution (i.e., moving objects, GPS trajectory data, etc.),

for example, generated through sensor networks, has clearly emerged as a new and popular research frontier in the field.

In addition to the many paper sessions, GIScience2010 also offered four peer-reviewed workshops, and for the first time also four tutorials one day before the main conference. These pre-conference events were intended as complementary opportunities to additionally facilitate dialogue across disciplinary boundaries. Another novelty included a doctoral colloquium after the main conference. Two keynote speakers, two poster sessions, an expert panel, a capstone talk, as well as social events rounded off the stimulating GIScience2010 conference activities. The conference dinner was held on top of Zurich's "house mountain," at the Üetliberg (800m), reachable on foot or by train, with stunning views of the entire city, Lake Zurich and surroundings, including the Swiss Alps.

Organizing a successful conference is not possible without the commitment, additional effort, and diligent help of many people we would like to thank: the international Program Committee for timely and thorough reviews; the sponsors and supporters for providing travel support for students and keynote speakers, for supplying materials at the conference, and for supporting social events. Furthermore, the organizers of the workshops, tutorials, and doctoral colloquium contributed an important part of the overall program. We would also like to thank the Conference Chair Robert Weibel and the Zurich organizing crew for all the hard work behind the scenes. Our special thanks go to Annica Mandola and Lisa Büschlen at the Department of Geography of the University of Zurich, who efficiently handled all administrative matters, and Ross Purves, who not only designed and maintained the conference website, but also swiftly managed the EasyChair conference management system, including respective digital communication and outreach matters. Finally, we would like to thank the most important people at any conference – those who came to present and discuss their work, and who by so doing demonstrated the continuing strength of GIScience as a discipline in its own right.

June 2010

Sara Irina Fabrikant
Tumasch Reichenbacher
Marc van Kreveld
Christoph Schlieder

Organization

Program Chair

Sara Irina Fabrikant	University of Zurich, Switzerland

Program Co-chairs

Tumasch Reichenbacher	University of Zurich, Switzerland
Marc van Kreveld	University of Utrecht, The Netherlands
Christoph Schlieder	University of Bamberg, Germany

General Chair

Robert Weibel	University of Zurich, Switzerland

Workshop Chair

Ross S. Purves	University of Zurich, Switzerland

Program Committee

Dave Abel	CSIRO, Australia
Pragya Agarwal	Lancaster University, UK
Ola Ahlqvist	Ohio State University, USA
Luc Anselin	Arizona State University, USA
Walid Aref	Purdue University, USA
Marc Armstrong	University of Iowa, USA
Kate Beard-Tisdale	University of Maine, USA
Scott Bell	University of Saskatchewan, Canada
Itzhak Benenson	Tel Aviv University, Israel
Michela Bertolotto	University College Dublin, Ireland
Ling Bian	University at Buffalo, USA
Thomas Bittner	University at Buffalo, USA
Claus Brenner	Leibniz University Hannover, Germany
Dan Brown	University of Michigan, USA
Dirk Burghardt	Techical University of Dresden, Germany
Gilberto Camara	INPE, Brazil
Nicholas Chrisman	Laval University, Canada
Christophe Claramunt	NARI, France
Helen Couclelis	UC Santa Barbara, USA

Tom Cova	University of Utah, USA
Isabel Cruz	University of Illinois at Chicago, USA
Leila de Floriani	University of Genova, Italy
Martin Dodge	University of Manchester, UK
Jürgen Döllner	Hasso Plattner Institut, Potsdam, Germany
Matt Duckham	University of Melbourne, Australia
Jason Dykes	City University London, UK
Max Egenhofer	University of Maine, USA
Sara Irina Fabrikant	University of Zurich, Switzerland
Peter Fisher	University of Leicester, UK
Stewart Fotheringham	National Centre for Geocomputation, Ireland
Andrew Frank	Vienna University of Technology, Austria
Christian Freksa	University of Bremen, Germany
Mark Gahegan	University of Auckland, New Zealand
Michael Goodchild	UC Santa Barbara, USA
Joachim Gudmundsson	NICTA, Sydney, Australia
Lars Harrie	Lund University, Sweden
Francis Harvey	University of Minnesota, USA
Mary Hegarty	UC Santa Barbara, USA
Gerard Heuvelink	Wageningen University, The Netherlands
Stephen Hirtle	University of Pittsburgh, USA
Kathleen Stewart Hornsby	University of Iowa, USA
Piotr Jankowski	San Diego State University, USA
Christopher Jones	Cardiff University, UK
Marinos Kavouras	NTUA, Greece
Menno-Jan Kraak	ITC, The Netherlands
Marc van Kreveld	Utrecht University, The Netherlands
Werner Kuhn	University of Münster, Germany
Mei-Po Kwan	Ohio State University, USA
Phaedon Kyriakidis	UC Santa Barbara, USA
Brian Lees	Australian Defence Force Academy, Australia
Paul Longley	University College London, UK
Alan MacEachren	Pennsylvania State University, USA
William Mackaness	University of Edinburgh, UK
David Mark	University at Buffalo, USA
Harvey Miller	University of Utah, USA
Daniel R. Montello	UC Santa Barbara, USA
Nora Newcombe	Temple University, USA
David O'Sullivan	University of Auckland, New Zealand
Atsuyuki Okabe	University of Tokyo, Japan
Harlan Onsrud	University of Maine, USA
Dimitris Papadias	UST, Hong Kong, SAR China
Ross Purves	University of Zurich, Switzerland
Jonathan Raper	City University London, UK
Martin Raubal	UC Santa Barbara, USA
Tumasch Reichenbacher	University of Zurich, Switzerland

Femke Reitsma	University of Canterbury, New Zealand
Andrea Rodriguez	Universidad de Concepción, Chile
Anne Ruas	Institut Géographique National, France
Christoph Schlieder	University of Bamberg, Germany
Nadine Schuurman	Simon Fraser University, Canada
Monika Sester	Leibniz University Hannover, Germany
Wenzhong Shi	Hong Kong Polytechnic, SAR China
Takeshi Shirabe	Vienna University of Technology, Austria
Ashton Shortridge	Michigan State University, USA
Jack Snoeyink	University of North Carolina, USA
Bettina Speckmann	Eindhoven University of Technology, The Netherlands
Paul Sutton	University of Denver, USA
Sabine Timpf	University of Augsburg, Germany
Ming Tsou	San Diego State University, USA
Monica Wachowicz	Wageningen University, The Netherlands
Robert Weibel	University of Zurich, Switzerland
Stephan Winter	University of Melbourne, Australia
Jo Wood	City University London, UK
Michael Worboys	University of Maine, USA
May Yuan	University of Oklahoma, USA

Additional Reviewers

Alexander Klippel
Alexei Pozdnoukhov
Anna-Katharina Lautenschütz
Arzu Çöltekin
Frank Ostermann
Matthew Niblett
Patrick Laube
Stefan Steiniger
Stefano De Sabbata
Thora Tenbrink

Sponsoring Institutions

Table of Contents

A Conceptual Data Model for Trajectory Data Mining

Vania Bogorny[1], Carlos Alberto Heuser[2], and Luis Otavio Alvares[2]

[1] Depto de Informatica e Estatistica – Universidade Federal de Santa Catarina
Campus Universitario C.P. 476, Florianopolis, Brazil
vania@inf.ufsc.br
[2] Instituto de Informatica – Universidade Federal do Rio Grande do Sul
Av. Bento Goncalves 9500, Porto Alegre, Brazil
{heuser,alvares}@inf.ufrgs.br

Abstract. Data mining has become very popular in the last years, and it is well known that data preprocessing is the most effort and time consuming step in the discovery process. In part, it is because database designers do not think about data mining during the conceptual design of a database, therefore data are not prepared for mining. This problem increases for spatio-temporal data generated by mobile devices, which involve both space and time. In this paper we propose a novel solution to reduce the gap between databases and data mining in the domain of trajectories of moving objects, aiming to reduce the effort for data preprocessing. We propose a general framework for modeling trajectory patterns during the conceptual design of a database. The proposed framework is a result of several works including different data mining case studies and experiments performed by the authors on trajectory data modeling and trajectory data mining. It has been validated with a data mining query language implemented in PostGIS, that allows the user to create, instantiate and query trajectory data and trajectory patterns.

Keywords: conceptual model, data mining, trajectory data, trajectory patterns.

1 Introduction

Since its origin, database design has the purpose of modeling data for operational purposes only. However, with the globalization and information dissemination the need to extract more interesting information and knowledge from large amounts of data by far exceeds operational database models. This is specially true for trajectories of moving objects, which is a new kind of spatio-temporal data generated by mobile devices, that in the last few years have become very popular in daily life.

Knowledge Discovery in Databases (KDD), which is the technology to extract interesting and previously unknown patterns from data, gives support for strategic and decision making purposes, and it is an advanced step towards intelligent data analysis. Database designers, however, have not yet realized the need of thinking about data mining when designing a database. While on the one side

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 1–15, 2010.

database designers focus on operational functionalities only, on the other side the data analysts have a lot of work to preprocess large databases for data mining. This results in the *gap* that exists between databases and data mining, having several consequences. The most relevant is the preprocessing phase for knowledge discovery, where data have to be filtered, cleaned, and transformed for data mining algorithms. It has been stated that data preprocessing for data mining consumes between 60% and 80% of the whole efford required in the KDD process [1] for conventional data. This problem increases even further when dealing with spatial and spatio-temporal data, which are the focus of this paper.

Several attempts have been made to overcome the limitations of database systems over data mining tasks [2]. One well known approach is the development of data mining query languages for temporal data [3] and spatial data [4]. Most of these attempts propose extensions of SQL that might be difficult to achieve, and after the years have not been adopted by commercial DBMSs [5]. Another problem is that most of these works focus on the mining step itself, without addressing the most important and time consuming task in the discovery process: *data preprocessing* [1] [5] and *pattern post-processing*. The problems with SQL and commercial systems motivated interest in using other database languages. For instance, [6] proposed an algebra to integrate data and patterns.

Another problem for several data mining tasks is that data have to be preprocessed and transformed into different granularities, otherwise no patterns may be discovered [7]. This is specially true for trajectories of moving objects [8]. Both data preprocessing for data mining and granularity transformation are arduous tasks from the user's perspective, and these problems could be significantly reduced if data mining was foreseen during the conceptual database design.

In previous work we have shown that data mining should consider not only the data, but the schema of the data as well [9]. Conceptual schemas represent all relationships among the data that are well known by the database designer, specially those which represent integrity constraints. These relationships are uninteresting for knowledge discovery, since they lead to the discovery of well known relationships among data. We have shown in [9] that all relationships with cardinality constraints *one..one* or *one..many* represented in the conceptual schema will generate association patterns that are well known and uninteresting. In [10] we have shown that the conceptual schema can be used to both visualize and represent spatio-temporal patterns of trajectories of moving objects. These examples show that the conceptual schema of a database should be considered during the KDD process.

Recently, Spaccapietra [11] proposed the first conceptual model for trajectories of moving objects. This model has been defined in the context of the European Project GeoPKDD [12], that has its focus on developing the science of trajectories of moving objects, from conceptual modeling, to databases, data warehouses and privacy-preserving data mining. This model is called *stops and moves*, and has been adopted by several approaches as a standard for *semantic trajectory data analysis*, as for instance, [13] [10] [14] [15] [16] [8].

In this paper we propose a novel solution to reduce the problem of data preprocessing for data mining, by reasoning and thinking about data mining during the

database conceptual design. We strongly believe that it is time to consider data mining already during the conceptual modeling phase of a database. Therefore, we propose a conceptual framework for modeling trajectory data and trajectory patterns. The model foresees three main types of trajectory patterns, previously defined in [8]: *frequent patterns, sequential patterns*, and *association rules*. The proposed framework is an extension of the conceptual model for trajectories proposed by [11], in order to support *semantic trajectory pattern mining*.

The proposed data mining modeling framework is inspired on experiments performed on real trajectory data, by two of the authors, in previous works as, for instance, [13] [16] [17] [8] [15]. From this set of results, which cannot be detailed because of space limitations, we propose a conceptual framework for trajectory conceptual modeling and mining, which can be used as a design pattern for semantic trajectory conceptual modeling.

The reminder of the paper is organized as follows: Section 2 introduces the basic concepts on trajectory conceptual modeling and trajectory data mining. Section 3 presents the proposed framework, which is evaluated with query examples in Section 4. Section 5 concludes the paper and suggests directions of future research.

2 Trajectory Conceptual Modeling and Mining

Several data models have been proposed for efficiently querying moving objects [18,19,20]. In [18] the main focus relies on the geometric properties of trajectories, while [20] considered semantics and background geographic information. In [20] a semantic model for trajectories has been proposed as well as relationships of trajectories with geographic and environment information. This model, however, is restricted to a specific application domain, and trajectory relationships are related to vehicles and roads. In [19] trajectories are modeled over road networks. This kind of approach is limited to the road network, while in several applications there exits no network, such as animal migration, recreational activities in a park, the behavior of people in a shopping center, etc.

SECONDO and HERMEs are spatio-temporal DBMS prototypes in which a lot of effort has been made for representing trajectories of moving objects, creating new data types and operations to manipulate the spatio-temporal properties of trajectories. However, very little has been done for trajectories from the application point of view. To overcome these problems, a new concept has been introduced for reasoning over trajectories from a semantic point of view. This concept is called *stops* and *moves* [11].

2.1 The Model of Stops and Moves of Trajectories

Spaccapietra [11] has introduced the first conceptual model for trajectories, which is an evolution of the spatio-temporal well known model *MADS* [21] to support trajectories. This model is application independent, and the user may add the semantics he is interested in, according to his application scenario. The semantic issue is very important because trajectory data are normally available

as sample points, without any semantics. Figure 1 shows an example of a trajectory sample (1) and a semantic trajectory (2). The *semantic trajectory* has associated the important places where the moving object has visited (the stops).

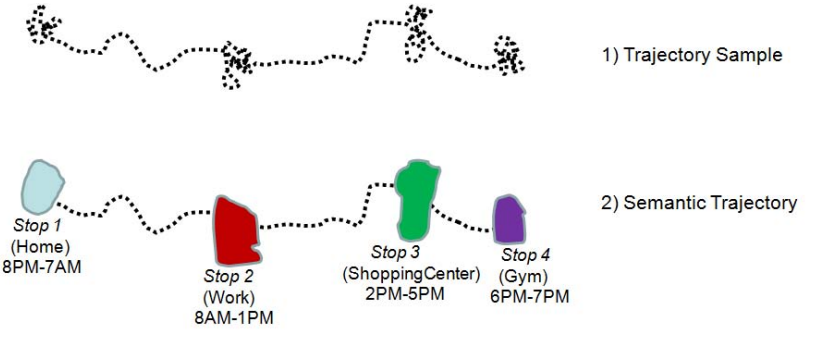


Fig. 1. Trajectory Sample (1) and Semantic Trajectory (2)

Considering trajectories from a semantic point of view, Figure 2 shows the model proposed by Spaccapietra, where a trajectory is the user defined record of the evolution of the position of an object that is moving in space during a given time interval, in order to achieve a given goal. In this model, a *Trajectory* belongs to an object (TravellingOT = Travelling Object Type). A trajectory is composed by a set of *stops* and *moves*. A *stop* (BES) is an important part of the trajectory from the application point of view where the moving object has stayed for a minimal amount of time. A *move* is a part of the trajectory between two consecutive stops. These concepts have been formally defined in [13].

BES represents the *Begin* and *End* of a trajectory or a *Stop*. Both stops and moves are related to a spatial object type (SpatialOT1 and SpatialOT2)), standardized by the Open GIS Consortium as *spatial feature type*.

A spatial feature type represents any spatial object in a set of spatial objects, in which a stop occurs. For instance, in a bird migration scenario, a stop can be at a specific country where birds are feeding. In a tourism application, a stop can be in a shopping center, a restaurant or a monument. In a traffic management application, a stop can be a traffic light, a roundabout, a parking place. Figure 1 shows an example of a trajectory sample (1) and a semantic trajectory (2), where the *semantic trajectory* has four stops: at home, at work, at a shopping center and at a gym. The moving points between the stops are the *moves*.

2.2 Semantic Trajectory Pattern Mining: Basic Concepts

In this section we describe the three data mining tasks supported by our model, considering the concept of stops and moves: trajectory frequent patterns, trajectory sequential patterns and trajectory association rules. These tasks are already supported and implemented in the first semantic trajectory data mining query

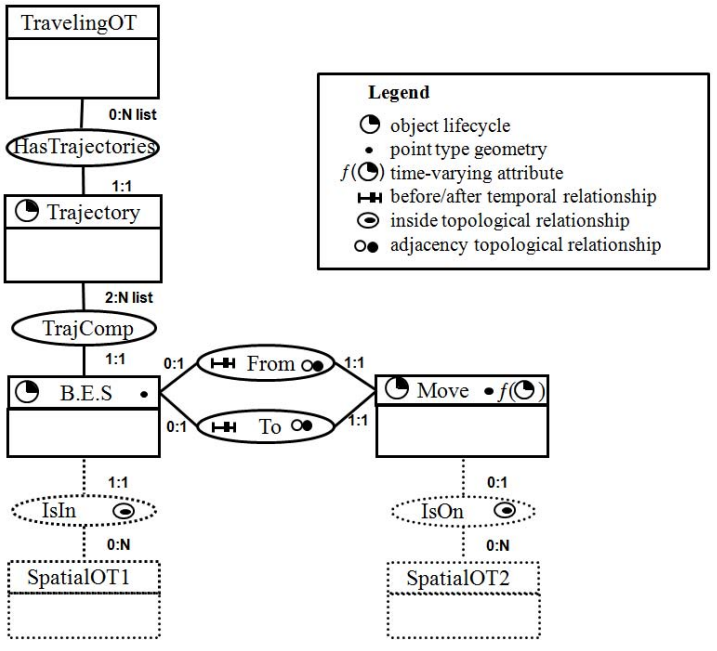


Fig. 2. The Model of stops and moves of trajectories [11]

language, named ST-DMQL, which we have introduced in [8]. First we define what is a semantic trajectory.

Definition 1. *Semantic Trajectory. A* Semantic Trajectory S *is a finite sequence* $\langle I_1, I_2, ..., I_n \rangle$ *where* I_k *is a stop or a move.*

Mining Trajectory *Frequent* Patterns. Frequent semantic trajectory patterns represent a set of items (stops or moves) that occur in a set of trajectories with support higher than a user defined minimum support (*minSup*).

Definition 2. *Support. Let* T *be a set of semantic trajectories. The support of a set of items* $X = \{x_1, x_2, ..., x_n\}$ *with respect to* T *is defined as the fraction of the trajectories in* T *that contain* X*, and is denoted as* $s(X)$*.*

A trajectory frequent pattern is defined as follows:

Definition 3. *Trajectory Frequent Pattern. Let* T *be a set of semantic trajectories. A set of items* $X = \{x_1, x_2, \ldots, x_n\}$ *is a trajectory frequent pattern with respect to* T *and minSup if* $s(X) \geq minSup$*.*

An example of trajectory frequent pattern is, for instance: $\{ReligiousPlace_{[weekend]}, Restaurant_{[weekend]}\}$ (s=0.07). This pattern expresses that 7% (support) of the trajectories stop at both religious places and restaurants on weekends. This pattern is a set with two items: $\{\text{ReligiousPlace}_{[weekend]}, \text{Restaurant}_{[weekend]}\}$. ReligiousPlace and Restaurant correspond to the spatial part of each item, while *weekend*

represents the time dimension of the item, i.e., when the moving objects stayed at these places. More details about *items* are presented in section 3.1.

Mining Trajectory *Sequential* Patterns. The sequential pattern mining technique differs from frequent patterns in the order as the items occur in a trajectory. In sequential patterns the order as the items occur plays the essential role. It corresponds to the relative order between items in the trajectories, but not necessarily the absolute order of the items. A Sequential trajectory pattern is a sequence of movements between items that have support higher than minimum support.

Definition 4. *Trajectory Sequential Pattern. Let T be a set of semantic trajectories. A sequence of items $X = \langle x_1, x_2, \ldots, x_n \rangle$, ordered in time, is a trajectory sequential pattern with respect to T and $minSup$ if $s(X) \geq minSup$.*

An example of sequential pattern is: $Work_{[morning]}$, $ShoppingCenter_{[afternoon]}$, $Gym_{[afternoon]}$ *(s=0.08%).* This pattern expresses that these three items occur in this temporal relative order in 8% of the trajectories.

Mining Trajectory Association Rules. Given an implication of the form $X \Rightarrow Y$ (s) (c), where X and Y are disjoint sets of items, the support s of the rule $X \Rightarrow Y$ is given as $s(X \cup Y)$. The confidence c is given as $s(X \cup Y)/s(X)$.

Definition 5. *Trajectory Association Rule. Let T be a set of semantic trajectories and X and Y be disjoint sets of items. An implication of the form $X \Rightarrow Y$ is a trajectory association rule with respect to T, $minSup$ and $minConf$ if $s(X \cup Y) \geq minSup$ and $s(X \cup Y)/s(X) \geq minConf$.*

Trajectory association rules represent associations among items. Considering our example of semantic trajectory shown in Figure 1 (2) and considering that a minimum number of trajectories would go from *home* to *work* and from there to a *Gym*, we could have an association rule such as:

$$Home_{[night]}\, Work_{[morning]} \Rightarrow Gym_{[afternoon]}\ \text{(s=0.10) (c=0.50)}$$

This rule expresses that in 10% of the trajectories the moving object has stayed at home at night, at a working place in the morning, and at a gym in the afternoon. Furthermore, among the trajectories that stop at home at night and at a working place in the morning, 50% also stop at a gym in the afternoon.

3 From Trajectory Conceptual Modeling to Data Mining

The model proposed in this paper is an extension of the model of stops and moves proposed by Spaccapietra to deal with trajectories from a semantic point of view. Therefore, our model is for mining *semantic trajectory patterns.* Figure 3 shows the proposed framework, represented as a UML class diagram. We represent the spatial attributes by the attribute *the_geom*, instead of stereotypes, and the time dimension by attributes. The classes in dark gray represent the data mining concepts, which are the main contributions of this paper. Including these concepts,

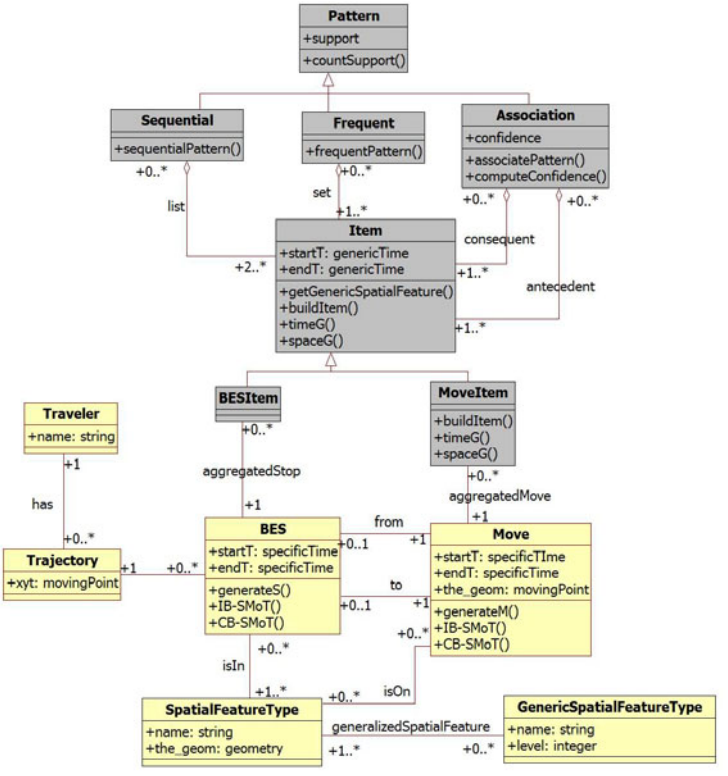


Fig. 3. A framework for Modeling Trajectory Data and Trajectory Patterns

that will be explained later in this section, we first extend the concept of *stops and moves* with two methods to automatically instantiate stops and moves, IB-SMoT and CB-SMoT, implemented and validated respectively in [13] and [16]. The method *generateS* computes stops based on one of the previous methods, while *generateM* generates the moves. These methods are defined, respectively, in the classes *BES* (Begin and End Stop) and *Move*. The model can easily be extended to support different methods to instantiate either stops or moves.

It is important to notice that in the original model of stops and moves the BES entity includes the begin and the end of the trajectory, while in our framework this entity corresponds to stops only. Therefore, the minimal cardinality between the entities Trajectory and BES has been changed from two to zero.

The attributes *startT* and *endT* represent the time in which Stops (BES) and Moves started and ended. They represent a specific time (e.g. 10:01). These attributes will be automatically instantiated by the methods that compute stops or moves (IB-SMoT and CB-SMoT). Both methods are supported by the first Semantic Trajectory Data Mining Query Language (ST-DMQL) [8], implemented in PostGIS (spatial database extension for PostgreSQL) and Weka[22], in order to instantiate the proposed model.

Patterns can be extracted from both stops and moves, but not directly. Very rarely patterns will be obtained directly from the data without aggregation. For instance, it is almost impossible that two or more moving objects would stop at *IBIS Hotel* exactly at the same time interval (e.g. 8:01 to 9:17). Patterns have to be extracted from data at aggregated levels. Therefore, we introduce the concept of *item*, which has been originally introduced in [8].

An *item* is the element that will be considered in the mining process, i.e., it is the element that will compose the pattern, which in trajectories is an aggregated stop or move. We call this element *item* because this term is well known in the classical data mining literature. In the context of trajectories, an *item* represents a set of information. While in transactional data mining an item is a single attribute like *milk* or *bread*, a trajectory item is a complex object that contains information about both *space* and *time*.

Stops and moves are defined and instantiated at the lowest granularity level, which we call as *feature instance* granularity. For data mining, these instances have to be generalized in order to discover patterns. Therefore, the classes *BESItem* and *MoveItem* specialize the *Item* and are respectivelly associated to one and only one stop or move.

Every *item* has the *time* information represented by the attributes *startT* and *endT*, and space information. The time information is a generalized time such as *weekend, weekday, month, year* and so on, while the time information at *BES* and *Moves* is a specific time, like 10:30. The *space* information of the item is the name and the type of the spatial feature related to the stop. The method *buildItem* will build the item, aggregating both space and time information using the methods *spaceG* and *timeG*, respectivelly. These three methods are redefined for the *MoveItem*, since an item composed by a move is a bit different from a stop. A *MoveItem* is build on two stops. These methods are detailed in Section 3.1.

The method *getGenericSpatialFeature* generalizes stops or moves related to spatial features through concept hierarchies. It will get from the class *GenericSpatialFeatureType* the respective level of a concept hierarchy to be considered for data mining.

The proposed model supports the three patterns described before: frequent patterns, sequential patterns, and association rules. The patterns have the attribute *support*, which is inherited by all three types of patterns, each of which is a subclass of pattern. Patterns are always extracted from items, therefore, patterns are aggregations of items, but with different meanings. A *sequential pattern* is a *list* of items ordered in time. A *frequent pattern* is a *set* of items, i.e., a set of stops or moves at any order. A *trajectory association rule* is an association pattern that is composed by a set of items that correspond to the *antecedent* of the rule and a set of disjoint items corresponding to the *consequent* of the same rule. The association pattern has an extra attribute, the *confidence* of the rule.

Notice in the model that for trajectory conceptual modeling the user may easily add new attributes, methods, or spatial feature types to stops (BES), according to the application requirements. The concepts corresponding to the pattern tasks, however, remain the same, i.e., for any application domain that is modeled using the extended model of stops and moves, the user will be able

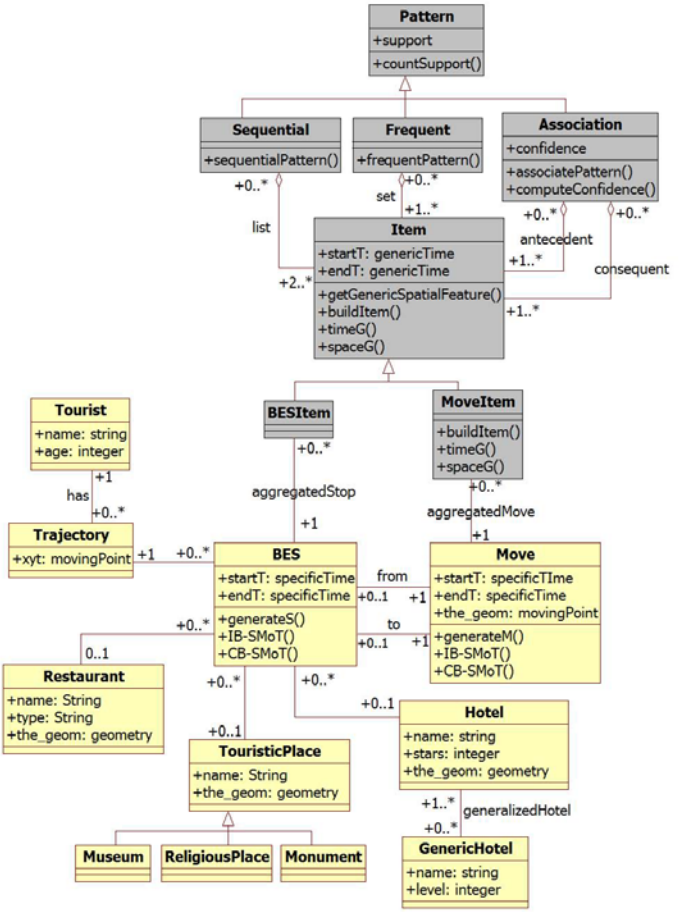


Fig. 4. A conceptual model for a trajectory tourism application

to extract frequent patterns, sequential patterns and association rules from trajectories without having to change the model.

Figure 4 shows an example of an instantiation of the proposed framework for the context of a tourism application. The travelers are tourists that have one or more trajectories. Every trajectory has zero or more stops (BES) that are interconnected by moves. In this example, a stop (BES) of a trajectory can occur at a hotel (Spatial Feature 1), restaurant (Spatial Feature 2), or a touristic place (Spatial Feature 3), which can be of type *museum*, *monument* or *religiousplace*. Hotel has a concept hierarchy associated (GenericHotel), which can be used for mining stops that occur at hotel at different granularities, such as 1 star, 2 stars, 3 stars and so on. Notice that any other relevant spatial feature that could generate a stop of a trajectory could be associated to *BES*, such as airport, shopping centers, and so on.

As can be seen in Figure 4, the main contribution of the proposed model is that for data mining, independently of the spatial features related to BES in the conceptual schema, the way as the item is build and the patterns are modeled remains standard. It is a *design pattern* for semantic trajectory pattern modeling.

3.1 Defining an Item

The *item* is the attribute that will be considered in the mining step, i.e., it is an element that will compose the pattern. In the context of trajectories, an *item* can be defined as: (i) *Name* - the name of the spatial feature where the moving object has passed and stayed for a minimal amount of time (e.g. Hotel, IbisHotel); (ii) *NameStart* - the name of the stop (spatial feature) with the time that the moving object has entered in the place (e.g. Hotel[morning], LouvreMuseum[10:00-12:00]); (iii) *NameEnd* - the name of the stop (spatial feature) and the time in which the moving object has left the place (e.g. Hotel[afternoon], LouvreMuseum[14:00-16:00]); and (iv) *NameStartEnd* - the name of the stop with the time period in which the moving object has respectively arrived and left the place (e.g. Hotel[morning][afternoon], LouvreMuseum[10:00-12:00][14:00-16:00]).

The item is build with the method *buildItem* defined in the class *Item*. It receives as an argument the *itemType*, which defines which space and time information will be considered in the item. According to the application, the time may be aggregated in different levels. In a traffic management application, it might be interesting to define rush hour at intervals like [7:00-9:00], [12:00-14:00], and [17:00-19:00]. In a tourism application, it might be interesting to aggregate time in [morning], [afternoon], and [evening]. The same is true for the space dimension. For instance, one may want to extract patterns at instance level (e.g. Ibis Hotel) or at feature type level (e.g. Hotel). In the following section we describe how the methods *spaceG* and *timeG*, defined in the class *Item*, work.

3.2 Time and Space Granularity (timeG and spaceG)

The time granularity is defined by the method *timeG*. This method generalizes a timestamp into different granularities, according to the user's interest. More specifically, it converts a timestamp (e.g. 08:12)into a semantic discretized *label* as [07:00-09:00].

The method *timeG* is very powerful because it allows the user to discretize the time dimension in several granularities. Please see [8] for details.

The language ST-DMQL that implements *timeG* provides some pre-defined time granularities, which are: WEEKEND-WEEKDAY, YEAR, MONTH, SEASON, and DAY-OF-THE-WEEK. For these specific granularities the user can use the parameter *pre_defined_TG* for the function *timeG*. This will automatically transform the time dimension of either stops or moves in the specified granularity. If the user is interested in specific time intervals, then he can decide for the parameter *user_defined_TG*, where he can specify a given time interval and choose a label, like for instance, *14:00-18:00* as *afternoon*.

The *space* granularity of either stops or moves is managed by the method *spaceG*. Two granularity levels can be automatically generated, without the need of any specification of background knowledge like concept hierarchies or ontologies. These granularities are called *feature instance* (e.g. Centrum Hotel) and *feature type* (e.g. Hotel). With the method *spaceG* the user can, for instance, set as default the granularity *type* or *instance*, and specify intermediate granularity levels defined in a concept hierarchy for some specific spatial feature types. Granularity levels different from *feature type* and *feature instance* are obtained from the class *GenericSpatialFeatureType*, with the method *getGenericSpatialFeatureType*, which has as a parameter the name of the generic spatial feature to be retrieved(e.g. Hotel). To better understand the *spaceG* function please see [8].

4 Evaluating the Proposed Model

To evaluate the proposed model we use the semantic trajectory data mining query language [8] implemented in PostGIS and Weka to both instantiate and query trajectory patterns. To facilitate the comprehension of the examples presented in this Section, we show how the conceptual model can be seen in a logic level. The relations of stops (that correspond to BES in the conceptual model) and moves have the following schema:

```
STOP (Tid integer, Sid integer, SFTname string,
     SFid integer, startT timestamp, endT timestamp)

MOVE (Tid integer, Mid integer, SFT1name string,
      SF1id integer, SFT2name string, SF2id integer,
      startT timestamp, endT timestamp, the_move geometry)
```

The attribute *SFTname* corresponds to the name of the spatial feature type and the attribute *SFid* is the identification of the instance (e.g. Ibis) of the spatial feature type (e.g. Hotel) in which the moving object has stopped. The attributes *SFT1name* and *SF1id* correspond to the stop in which a move starts, while the attributes *SFT2name* and *SF2id* represent the stop where the move finishes.

The relation which stores frequent patterns and sequential patterns has the following schema:

```
frequentPattern/
sequentialPattern (Pid integer, pattern itemSetType, support real)
```

The attributes are respectively the identifier of the pattern, which is a sequential number, the pattern itself, which we represent and store as a *nested relation* [23], and the support of the pattern. The type *itemsettype*, which represents a nested relation, is defined by the following structure:

```
itemSetType (SFT1name string, SF1id integer, SFT2name string,
  SF2id integer, startT string, endT string)
```

The pattern schema is similar to the structure of the move relation, because it stores both patterns of stops and patterns of moves. The main difference is

that instances of a pattern represent aggregated information of stops and moves, and not the instances of stops and moves.

When the pattern is a move pattern, then *SFT1name, SFT1id* and *SFTname2, SFT2id* are two stops that form a move pattern. If the pattern is generated from stops, then *SFTname2, SFT2id* will be null.

Concerning trajectory association rules, the association pattern relation has the following structure:

```
associatePattern (Pid integer, antecedent itemSetType,
 consequent itemSetType, support real, confidence real)
```

The attributes are respectively the identifier of the rule, the antecedent of the rule, the consequent, the support, and the confidence. In this type of pattern both antecedent and consequent are stored as nested relations, and therefore both have the structure of the *itemSetType*. These structure is what allows quering patterns just as data, as will be shown in Section 4.2.

4.1 Instantiating Patterns

Considering the context of a tourism application presented in Figure 4, let us suppose that the user wants to know the types of places most frequently visited by tourists on weekdays and weekends. This question can be answered by the following simple query to extract frequent patterns:

```
SELECT frequentPattern (itemType=NameStart, timeG=WEEKEND-WEEKDAY,
                        spaceG=[type,GenericHotel=1], minsup=0.15)
  FROM stop
```

In this query, the item that composes the pattern will contain the name of the stop and the time in which the stop started. This is expressed by the parameter *itemType=NameStart*. The time granularity (timeG) is weekday and weekend, while the granularity of space will be at the feature type level for all stops, except for those that occur at hotels. For hotels, a concept hierarchy *GenericHotel* is used at the granularity level 1, and therefore touristic stops at hotels will be aggregated by types like, for instance, 2StarsHotel, 3StarsHotel, etc...

The query will generate patterns over stops that appear in at least 15% of the trajectories, and the output of this query will be frequent patterns as, for instance:

$\{4StarsHotel_{[weekend]}, Museum_{[weekend]}, Restaurant_{[weekend]}\}$ (s=0.16)

Now let us suppose that the user wants to know the sequences of moves that occur most frequently in the morning and in the evening. This question can be answered by the following query:

```
SELECT sequentialPattern (itemType=NameEnd, timeG=[8:00-12:00 AS morning,
                    18:00-23:00 AS evening],spaceG=instance, minsup=0.03)
  FROM move
```

In this query a move has to occur in at least 3% of the trajectories in order to generate a pattern. The pattern will contain the name of the move and the time

that the move finishes, and time will be aggregated in morning and evening. In this data mining query there is no aggregation on the name of the stop, i.e., the patterns will contain the places where the moves happened. A sequential pattern that will be generated by this query may be the following:

$$\{IbisHotel\text{-}NotreDame_{[morning]},\ EiffelTower\text{-}IbisHotel_{[evening]}\}\ (s{=}0.04)$$

Considering now a traffic management application (the conceptual model is not shown because of space limitation), where traffic jams (very low speed regions) are the important places in trajectories. The stops have been instantiated with the method CB-SMoT and the user, for instance, may ask a question like: which are the relationships among stops (traffic jams) in a city at different time periods during rush hours? This question may be answered by the following query:

```
SELECT associatePattern(itemType=NameStart, spaceG=instance,
                        timeG=[07:00-08:00, 08:01-09:00,
                        16:00-17:00,17:01-18:00,18:01-19:00],
                        minsup=0.1, minconf=0.40)  FROM stop
```

This query will generate association rules that have at least 10% support and at least 40% confidence. Each *item* in the rule, either in the antecedent or the consequent, will contain the name of the stop and the time period in which the stop starts (itemType=NameStart). The stop granularity will be at the feature instance level for all stops (spaceG=instance), and the time are specific hours within the rush hours (timeG). From this query the user may obtain association rules like, for instance:

$$AvGrandArmee_{[07:00-08:00]} \Rightarrow AvChampsElisees_{[08:01-09:00]}\ (s{=}0.06)\ (c{=}0.40)$$

4.2 Querying Trajectory Patterns

Since the patterns are stored as database relations, any query can be performed. Any filter or constraint may be applied directly over the patterns. For instance, considering the tourism application, let us suppose that the user is interested in association patterns which have *weekend* as the time dimension in the antecedent of the rule. A query like the following can be performed:

```
SELECT *
  FROM associatePattern
 WHERE antecedent.startT='weekend' or antecedent.endT='weekend'
```

The power of storing the discovered patterns for further analysis allows complex queries, where the geometry of either stops or moves can be used to filter the patterns. For instance, suppose the user is interested in how many moves that cross the bridge Pont Neuf are contained in sequential patterns. This query will join three relations: sequential patterns, moves, and bridge, as follows:

```
SELECT count(m.*)
  FROM sequentialPattern s, bridge b, move m
 WHERE s.pattern.SFT1name=m.SFT1name AND s.pattern.SF1id=m.SF1id AND
       s.pattern.SFT2name=m.SFT2name AND s.pattern.SF2id=m.SF2id AND
       b.name='Pont Neuf' AND intersects(m.the_geom,b.the_geom)
```

5 Conclusions

In this paper we presented a general framework for conceptually modeling trajectory patterns. The proposed model is an extension of the conceptual model proposed by Spaccapietra for modeling trajectories of moving objects from a semantic point of view. We extend this model to support semantic trajectory patterns, that are extracted from aggregated stops and moves of trajectories. The model provides to the user semantic trajectory sequential patterns, trajectory association rules and trajectory frequent patterns. The proposed model can be implemented using the ST-DMQL, a semantic trajectory data mining query language developed for defining, mining and querying trajectory patterns at different space and time granularity levels. The proposed model significantly reduces the preprocessing tasks for data mining, which normally are the most effort and time consuming tasks. Furthermore, it facilitates pattern filtering in postprocessing. The proposed framework is a result of several works including different data mining case studies and experiments performed by two of the authors on trajectory data modeling and trajectory data mining. As future works we are investigating the extension of the model to support trajectory clustering and trajectory classification.

Acknowledgements

This work was supported by the Brazilian agencies CAPES (Prodoc Program), FAPESC (CP005/2009) and CNPq (550891/2007-2, 573871/2008-6 and 307588/2008-4).

References

1. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann, San Francisco (1999)
2. Wang, H., Zaniolo, C.: Atlas: A native extension of sql for data mining. In: Barbará, D., Kamath, C. (eds.) SDM. SIAM, Philadelphia (2003)
3. Chen, C.X., Kong, J., Zaniolo, C.: Design and implementation of a temporal extension of sql. In: Dayal, U., Ramamritham, K., Vijayaraman, T.M. (eds.) ICDE, pp. 689–691. IEEE Computer Society, Los Alamitos (2003)
4. Malerba, D., Appice, A., Ceci, M.: A data mining query language for knowledge discovery in a geographical information system. In: Meo, R., Lanzi, P.L., Klemettinen, M. (eds.) Database Support for Data Mining Applications. LNCS (LNAI), vol. 2682, pp. 95–116. Springer, Heidelberg (2004)
5. Boulicaut, J.F., Masson, C.: Data mining query languages. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 715–727. Springer, Heidelberg (2005)
6. Calders, T., Lakshmanan, L.V.S., Ng, R.T., Paredaens, J.: Expressive power of an algebra for data mining. ACM Transactions on Database Systems 31(4), 1169–1214 (2006)
7. Han, J.: Mining knowledge at multiple concept levels. In: CIKM, pp. 19–24. ACM Press, New York (1995)
8. Bogorny, V., Kuijpers, B., Alvares, L.O.: St-dmql: a semantic trajectory data mining query language. International Journal of Geographical Information Science 23(10), 1245–1276 (2009)

9. Bogorny, V., Kuijpers, B., Alvares, L.O.: Reducing uninteresting spatial association rules in geographic databases using background knowledge: a summary of results. International Journal of Geographical Information Science 22, 361–386 (2008)
10. Alvares, L.O., Bogorny, V., de Macedo, J.F., Moelans, B., Spaccapietra, S.: Dynamic modeling of trajectory patterns using data mining and reverse engineering. In: Twenty-Sixth International Conference on Conceptual Modeling - ER2007 - Tutorials, Posters, Panels and Industrial Contributions, November 2007. CRPIT, vol. 83, pp. 149–154 (2007)
11. Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., Vangenot, C.: A conceptual view on trajectories. Data and Knowledge Engineering 65(1), 126–146 (2008)
12. GeoPKDD, P (2006), http://wwww.geopkdd.eu
13. Alvares, L.O., Bogorny, V., Kuijpers, B., de Macedo, J.A.F., Moelans, B., Vaisman, A.: A model for enriching trajectories with semantic geographical information. In: ACM-GIS, pp. 162–169. ACM Press, New York (2007)
14. Baglioni, M., de Macêdo, J.A.F., Renso, C., Wachowicz, M.: An ontology-based approach for the semantic modeling and reasoning on trajectories. In: ER Workshops, pp. 344–353 (2008)
15. Bogorny, V., Wachowicz, M.: A framework for context-aware trajectory data mining. In: Cao, L., Yu, P.S., Zhang, C., Zhang, H. (eds.) Data Mining for Business Applications, pp. 225–240. Springer, Heidelberg (2008)
16. Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O.: A clustering-based approach for discovering interesting places in trajectories. In: ACMSAC, pp. 863–868. ACM Press, New York (2008)
17. Alvares, L.O., Oliveira, G., Heuser, C.A., Bogorny, V.: A framework for trajectory data preprocessing for data mining. In: International Conference on Software Engineering and Knowledge Engineering, pp. 698–702 (2009)
18. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving objects databases: Issues and solutions. In: Rafanelli, M., Jarke, M. (eds.) SSDBM, pp. 111–122. IEEE Computer Society, Los Alamitos (1998)
19. Güting, R.H., de Almeida, V.T., Ding, Z.: Modeling and querying moving objects in networks. VLDB Journal 15(2), 165–190 (2006)
20. Brakatsoulas, S., Pfoser, D., Tryfona, N.: Modeling, storing, and mining moving object databases. In: IDEAS, pp. 68–77. IEEE Computer Society, Los Alamitos (2004)
21. Parent, C., Spaccapietra, S., Zimanyi, E.: Conceptual Modeling for Traditional and Spatio-Temporal Applications – The MADS Approach. Springer, Heidelberg (2006)
22. Frank, E., Hall, M.A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: Weka - a machine learning workbench for data mining. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 1305–1314. Springer, Heidelberg (2005)
23. Abiteboul, S., Schek, H.-J., Fischer, P.C. (eds.): NF2 1987. LNCS, vol. 361. Springer, Heidelberg (1989)

Time-Geographic Density Estimation for Moving Point Objects

Joni A. Downs

Department of Geography, University of South Florida, Tampa, FL, USA
downs@usf.edu

Abstract. This research presents a time-geographic method of density estimation for moving point objects. The approach integrates traditional kernel density estimation (KDE) with techniques of time geography to generate a continuous intensity surface that characterises the spatial distribution of a moving object over a fixed time frame. This task is accomplished by computing density estimates as a function of a geo-ellipse generated for each consecutive pair of control points in the object's space-time path and summing those values at each location in a manner similar to KDE. The main advantages of this approach are: (1) that positive intensities are only assigned to locations within a moving object's potential path area and (2) that it avoids arbitrary parameter selection as the amount of smoothing is controlled by the object's maximum potential velocity. The time-geographic density estimation technique is illustrated with a sample dataset, and a discussion of limitations and future work is provided.

Keywords: time geography, moving objects, density estimation, point pattern analysis.

1 Introduction

Methods of spatial point pattern analysis are widely used in GIScience to characterise the distribution of a set of objects or phenomenon that are represented as geometric points. In the statistical literature, point objects of interest are termed *events*, while all generic locations within a study region are referred to as *points* [1]. Point pattern analysis can take a variety of forms that range from the simple generation of a footprint delineating the area occupied by the events [2] to measuring whether the spatial pattern of events displays complete spatial randomness, clustering, or regularity [3] to more sophisticated methods of cluster analysis that identify and demarcate individual clusters of events [4].

One of the most popular methods of point pattern analysis applied in GIScience is kernel density estimation (KDE), which generates a smooth, continuous surface from a point pattern that represents spatial variation in the density of events [5]. KDE computes the spatial intensity at each point in the study region using a kernel function that weights the contribution of events based on their distance from each point. KDE is often applied as a form of 'hot spot' analysis where the goal is to identify locations where the spatial intensity of events is unusually high. For example, KDE has been

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 16–26, 2010.

used to identify hot spots of crime [6], disease [7], fatalities [8], traffic accidents [9], and lightning strikes [10]. Additionally KDE has been applied to characterise the spatial distribution of plants [11], animals [12], and people [13].

Despite the widespread usage of KDE across numerous disciplines, several authors have noted that the technique can perform poorly in some situations and for specific types of point patterns [14-16]. Point patterns with spatial distributions of events that conform to complex shapes can be particularly problematic for KDE to characterise accurately [17]. For the case of linearly arranged events, an adaptation of KDE to the network data model has proven a useful solution [18,19]. Another situation where KDE produces biased results is for point patterns generated by moving objects, such as those that document the movements of an individual pedestrian or animal over time [14-17]. One explanation for the poor performance of KDE in this case is that the assumption of independent events is violated when the point pattern is generated by a single object moving through space. Given this observation, a density estimation technique for moving point objects is needed to characterise these types of patterns more reliably.

This research introduces a method of density estimation for moving point objects where the events represent observed spatial locations for a single individual over time. The approach combines traditional density estimation with techniques of time geography to generate a continuous intensity surface that characterises the spatial distribution of the moving object over a fixed time frame. The resulting intensity surface represents a relative frequency distribution of the object's location; in other words it identifies particular locations where the object was most often located. Such a method may be useful for analysing the distribution of a variety of moving objects, such as pedestrians, animals, or ships.

The remainder of this paper is organised into the following sections. First, Section 2 provides a background detailing the mechanics of kernel density estimation for spatial point patterns. Then, Section 3 provides an overview of time geography and relevant mathematical notation that provides a foundation for this research. Section 4 details the new time-geographic density estimation technique, which is illustrated with a practical example. Finally, the paper closes with a discussion of results and guidance for future work in Section 5.

2 Kernel Density Estimation

Kernel density estimation is a well-known general data smoothing technique in statistics [5], with bivariate KDE often applied for spatial point pattern analysis [1]. In the latter context, the technique applies a kernel function over a point pattern to generate a continuous surface that quantifies the density of events at each point. The bivariate kernel density estimate f at any point x can be computed as:

$$\hat{f}(x) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{1}$$

where sample size n contains events $X_1, X_2, \ldots, X_n$ and $K(y)$ represents the kernel function that operates on the distance between each point and event given a specified

bandwidth h [5]. Any continuous, non-negative and radially symmetric kernel that integrates to 1 can be used, such as the Gaussian or Epanechnikov [20]. The choice of kernel is not particularly important, as most tend to produce similar patterns of smoothing. The degree of smoothing is controlled by the kernel's bandwidth, sometimes called a window width, which essentially specifies the distance within which events will contribute to the density estimate. Bandwidths can be chosen arbitrarily or with algorithms that attempt to find an optimal value. Additionally, the bandwidth can be fixed (constant) or adaptive (variable width, depending on the local density). It is also possible to specify different parameters for the x and y directions. Fig. 1 illustrates the application of KDE to a spatial point pattern, using a quartic kernel and a fixed bandwidth with equal x and y values.

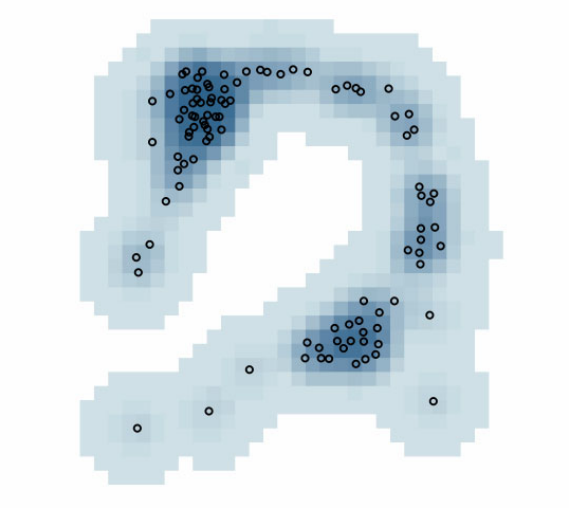


Fig. 1. KDE applied to a spatial point pattern

While KDE can often identify hotspots of events, as the figure above suggests, the technique has important limitations. One of its main disadvantages is its sensitivity to bandwidth selection, as different values produce dramatically different results [20]. Secondly, KDE has been shown to perform poorly for point patterns that result from moving point objects [18-19]. Since moving objects have both spatial and temporal components, the latter of which is often ignored in spatial point pattern analysis, a density estimation technique that incorporates time might overcome independence assumptions and offer a more reliable method of analysis. Time geography, as described in the next section, can provide a framework for this type of analysis.

3 Time Geography

First conceptualised by Hägerstrand [21], time geography provides a framework for modelling human behaviour and interaction given known observations of spatial and temporal activities along with constraints on movement capabilities. Miller's [22,23]

subsequent mathematical formulation of time geography provides the basis for analysing moving objects in GIScience. Here, only the fundamental concepts of time geography are reviewed in simplified form as they relate to the research at hand; for a more thorough documentation, readers should consult Miller's work in [22,23], which the notation below follows.

The trajectory of a moving object through space and time can be represented as space-time path $P(t)$, which consists of a set of ordered control points C and a set of path segments S. Each control point c_i is associated with a known time instance t_i and a coincident spatial location x_i. Control points immediately following c_i are denoted c_j. Thus, a set of n control points can be mathematically notated as:

$$C = \{c_S(t_S), \ldots, c_i(t_i), c_j(t_j), \ldots, c_E(t_E)\}, \tag{2}$$

where S and E indicate starting and ending points. For the purposes of this study, we assume the control points are random or regularly sampled locations from an object's movement trajectory over time. From the list of ordered control points, the individual segments of the space-time path can be estimated as:

$$S_{ij}(t) = \left(1 - \frac{t-t_i}{t_{j-t_i}}\right) x_i + \left(\frac{t-t_i}{t_{j-t_i}}\right) x_j \,. \tag{3}$$

Assuming a uniform velocity, the space-time path is approximated using straight-line segments to connect adjacent control points. A sample space-time path for a set of points, plotted in two-dimensional space, is shown in Fig. 2.

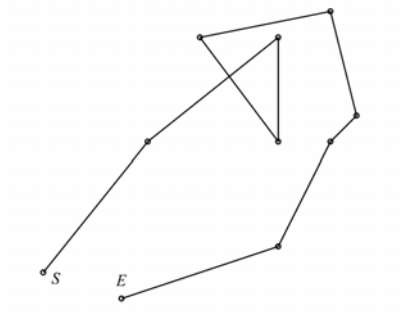


Fig. 2. A space-time path plotted in geographic space

From a space-time path for a moving object, it is possible to construct a space-time prism. The space-time prism delineates all locations for all time instances where the object could have been located, given spatial and temporal constraints imposed by the known control points along with the maximum possible velocity v of the object. Miller [23] provides a complete mathematical formulation for the space-time prism, although only the relevant portion of the computation is repeated here. Of interest is the geo-ellipse, or potential path area, for any two adjacent control points. The geo-ellipse delineates in geographic space all locations that are potentially reachable

during the time interval t_i to t_j; in other words it is a geometric footprint of the space-time prism for that time duration. The geo-ellipse g_{ij} can be formulated as:

$$g_{ij} = \{x \mid \|x - x_i\| + \|x_j - x\| \leq (t_j - t_i)v\}, \quad (4)$$

Where $\| \|$ represents a distance operator. Effectively, the geo-ellipse is an ellipse centered on the two control points, with its size and shape determined by the elapsed time and distance between control points and the maximum velocity value. Note that this formulation differs slightly from that of Miller [23] in that the activity parameter is omitted, as it is unnecessary because this study assumes the control points are recorded at time instances without any known durations of activity. Additionally, the maximum velocity parameter is assumed constant. A sample geo-ellipse for two control points is illustrated in Fig. 3.

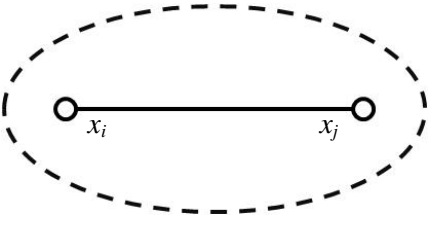


Fig. 3. A geo-ellipse constructed for two control points in a space-time path

The space-time path and prism, along with their extensions like the geo-ellipse, provide the foundation for geographic analysis of moving objects in GIScience. These fundamental concepts provide the basic analytical tools for a variety of analyses, such quantifying spatio-temporal interactions among objects or measuring the accessibility of objects to particular locations given movement constraints [24-26]. However, one area that remains largely unexplored in the context of time geography is density estimation. Winter [27] presented some initial exploratory work on a related topic of a generating a probability distribution for space-time prisms based on locational uncertainty, although it differs from the time-geographic density estimation method presented in the following section.

4 Time-Geographic Density Estimation

This section presents a new method of density estimation for moving point objects. The goal of the technique is to generate a continuous intensity surface that characterises the spatial distribution of an object over a fixed time frame. This can be accomplished by combining traditional kernel density estimation with techniques of time geography. This integration is straightforward if one considers the correspondence between the kernel in KDE and the geo-ellipse in time geography.

Recall that the kernel determines how much influence an event contributes to the density estimate at each location in space. A point that is close to a relatively large number of events will receive a high intensity. This is because kernels placed over each event overlap at that point, and the resulting intensity is large as the individual

values from each kernel are summed. While this approach may be sound for independent, stationary objects, it is problematic for moving objects for the following reason. Consider a simple scenario for a point pattern of two events which record the location of a moving object. A kernel can be placed over each of the two events for computing the kernel density estimates. Suppose a discrete kernel is used, such that the influence of each event only extends the distance of the bandwidth, as illustrated in Fig. 4. However, if one considers the geo-ellipse computed for those two events, clearly the kernel density estimates would be misleading, as intensities greater than zero would be assigned to areas where the object could not have possibly been located given time-geographic constraints. This observation may explain why other studies have shown KDE to perform poorly for moving point objects. While KDE is sensitive to the choice of bandwidth, the conclusions are unaffected as only the scale of the problem changes.

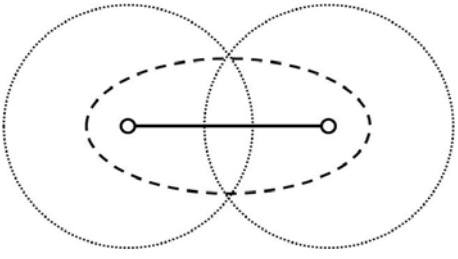


Fig. 4. Kernels (dotted lines) and geo-ellipse (dashed line) for two control points along a space-time path (solid line)

A potential solution to this problem is to compute the density estimates using the geo-ellipse as a surrogate for the kernels. Since the geo-ellipse delineates all possible locations for the object during the time duration between adjacent control points, it is intuitive that only locations within that area receive intensities greater than zero. When multiple events in a space-time path are recorded, the geo-ellipse would be generated for each pair of adjacent points and the density estimates would be summed at overlapping points as in traditional KDE (Fig.5). Densities within each geo-ellipse could be weighted according to distance similar to a kernel.

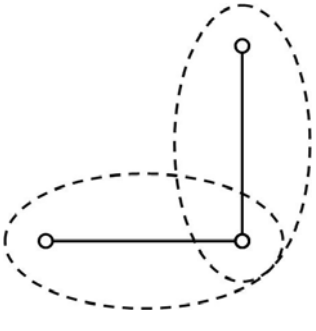


Fig. 5. Overlapping geo-ellipses for three control points along a space-time path

The time geographic density estimate f_t at any point x can be computed by adapting the original KDE formulation (eq. 1) using time geography:

$$\hat{f}_t(x) = \frac{1}{(n-1)[(t_E - t_S)v\ ]^2} \sum_{i=1}^{n-1} G\left(\frac{\left\|x - x_i\right\| + \left\|x_j - x\right\|}{(t_j - t_i)v} \right), \tag{5}$$

where the kernel is replaced with a function of the geo-ellipse G which is derived from eq. 4. For this function, the numerator sums the distance between a given point in space and the object's location at times i and j. The denominator computes the maximum distance the object could have travelled during that time interval given its maximum possible velocity; this value is equivalent to the bandwidth in KDE. Then the geo-ellipse function is applied to the ratio of these two distances in a similar manner to a traditional kernel function. Assuming the geo-ellipse function is discrete, such that no intensity is computed outside its boundaries, the ratio between those two distance values determines the intensity at the given point. If a uniform function is utilised, then the intensity will be smoothed evenly across the geo-ellipse; in other words, the likelihood of the object's occurrence is assumed equal at all locations within the geo-ellipse. If a distance-weighting function is used, such as a linear decay function, then locations nearer the known control points and along the approximated space-time path will receive higher intensities than those near the edge of the geo-ellipse. This approach is more reasonable if the object is unlikely to have been moving at maximum speed for the duration of the time interval. The remainder of the formula is updated in a similar manner, with the maximum travel distance replacing all instances of the KDE bandwidth. Additionally, n-1 replaces n, as there are n-1 geo-ellipses generated from n control points.

For illustration, the time-geographic density estimation technique is applied to a hypothetical set of 100 control points, the same point pattern of events used to illustrate KDE in Fig. 1. In this case, the control points are assumed to have been sampled at equal time intervals during the object's movement trajectory. Although the formulation can accommodate irregular time intervals, a regular spacing is chosen for simplicity. The space-time path of the object in geographic space is shown in Fig. 6. The width of each cell measures 1 distance unit while the maximum distance between control points is assumed to be 8 units. In other words, the maximum velocity of the object is 8 distance units/unit of time if there is a 1 unit time interval between control points. Fig. 6(a) illustrates the time-geographic density surface when a uniform function of the geo-ellipse is applied, while Fig. 6(b) shows the same for a linear decay function.

Both the uniform and linear decay geo-ellipse functions produce positive values within the same region but distribute the densities differently. The uniform function creates a more even distribution, while still highlighting areas with a higher density of control points. The linear decay function, on the other hand, computes relatively higher intensities to areas near control points and along the approximated space-time path.

Fig. 6 illustrates the advantage of the time-geographic approach compared to traditional KDE as previously illustrated in Fig. 1. While KDE is sensitive to parameter selection, the bandwidth was chosen to be as directly comparable to the new method as possible. When compared to the new method, it is clear that KDE produced positive

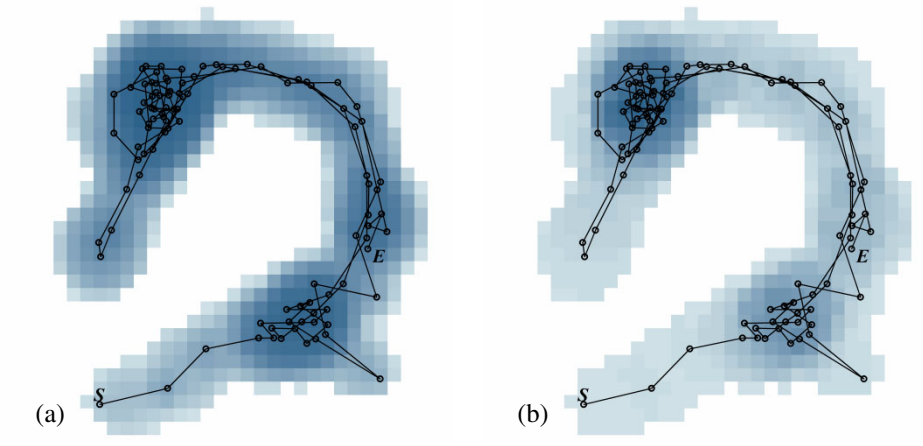


Fig. 6. Time-geographic density surfaces for a set of 100 control points using uniform (a) and linear decay (b) geo-ellipse functions. White areas indicate a zero density.

intensities in areas where the object could not have been located, given spatial and temporal constraints imposed by the control points and the maximum potential velocity. This observation would be true for relatively large bandwidths, while relatively small ones would produce a disjointed intensity surface that assigns density values of zero to some areas within geo-ellipses. Since kernels are always radially symmetric, even an 'optimal' bandwidth could not create a density surface that perfectly matches that produced using the time-geographic method. The sample application of the time-geographic density estimation technique suggests the method is capable of producing reasonable results for moving point objects.

5 Discussion and Conclusions

This research presents a new time-geographic method of density estimation for moving point objects. It integrates traditional methods of kernel density estimation with techniques of time geography to generate a continuous surface that characterises the spatial distribution of a moving object during a fixed time interval. There are two advantages to the time-geographic approach compared to traditional KDE. First, since the new method explicitly incorporates spatial, temporal, and maximum velocity constraints, no spatial intensity is assigned to locations where the moving object could not possibly have been located. This is in contrast with KDE which spreads the intensity in a radially symmetric fashion about each event. Second, unlike KDE, the time-geographic approach does not require the selection of an arbitrary bandwidth parameter. Its equivalent is determined from the maximum velocity of the moving object, a parameter that can potentially be determined accurately if the object's characteristics are well known. The results of this initial application to a sample set of control points highlight these advantages and demonstrate that the method can produce a reasonable representation of the underlying probability distribution of a moving object's location. This technique may be widely applicable for the analysis of moving objects in GIScience

and related disciplines. Such examples might include movements of fish, wildlife, people, vehicles, or ships.

However, there are some factors that might be considered before applying the technique in practice. First, while equation (5) directly accommodates any temporal sampling interval, the implication of this choice should be carefully considered. On one hand, there often is a trade-off between battery life and sampling frequency with GPS or other tracking devices [28], so this suggests wider intervals may be more practical. However, wider intervals generate a greater degree of uncertainty about an object's space-time path and thus produce larger geo-ellipses than those for relatively shorter time durations. If the sampling interval is too wide, then the approximated space-time path may not be accurate, and the corresponding geo-ellipses will be too large to be very informative. Consider Fig. 7 which computes the density surfaces with a linear geo-ellipse function for the same moving object used in sections 2 and 3.

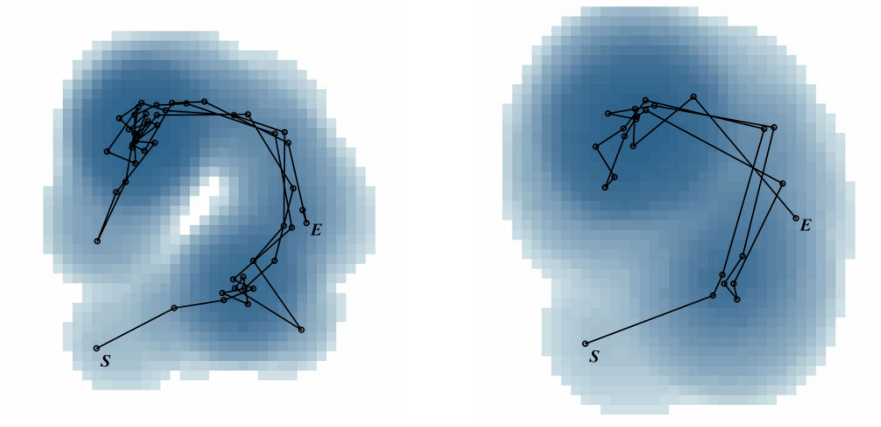


Fig. 7. Time-geographic density surfaces for a moving object at two temporal sampling frequencies: 2 time units (a) and 4 time units (b)

Fig. 7(a) illustrates the surface for every second control point sampled from the original distribution of 100 points, while (b) shows the same for every fourth control point. Clearly, wider sampling intervals generate less exact results. Future work may explore minimum sampling intervals that are necessary to accurately describe the distribution of particular types of moving objects.

In addition to interval width, the regularity of the temporal sampling scheme can affect the time-geographic density estimates. An irregular temporal sampling interval will create a density surface with variable precision; areas with control points sampled more frequently will be delineated more exactly than for those sampled less often. While this creates the most realistic picture of the uncertainty about an object's trajectory and probability distribution given the control points, a regular sampling interval may produce the most reliable results. Neutens et al. [29] discuss some implications about the effects of uncertainty in time geographic analysis, and future research may further explore the impacts of uncertainty—both temporal and spatial—in the context of density estimation.

There are two other limitations with the proposed formulation of the time-geographic density estimation technique that can be used to guide further research. As equation (5) is currently written, it assumes a constant maximum velocity. This may not be reasonable if the maximum velocity of an object changes over time or with respect to spatial location. Miller [23] provides a framework for incorporating non-uniform velocities into time geography, so future work may relax this assumption for the density estimation technique. Similarly, the formulation in this research assumes the Euclidian straight-line path between two control points is the most likely trajectory. However, this may not be valid if other path approximation methods are more accurate, for instance a curved pathway as demonstrated by [30], or if another distance metric is more appropriate, such as a network. Future research might explore the use of time-geographic density estimation in these contexts.

In conclusion, the time-geographic density estimation technique proposed in this research offers a new method for analysing moving point objects in GIScience. Specifically, the method generates a probability density surface for a moving object based on known control points and its maximum velocity. This research suggests the method may perform more accurately in some situations than other existing methods, particularly kernel density estimation, and may be useful for analysing a variety of moving objects. Future work can extend its applicability to a wider range of scenarios commonly encountered in GIScience.

Acknowledgements. This work was supported, in part, by the University of South Florida Internal Awards Program under Grant No. R071476.

References

1. Cressie, N.: Statistics for Spatial Data. Wiley, New York (1993)
2. Galton, A., Duckham, M.: What is the region occupied by a set of points? In: Raubal, M., Miller, H.J., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2006. LNCS, vol. 4197, pp. 91–98. Springer, Heidelberg (2006)
3. Diggle, P.: Statistical Analysis of Spatial Point Patterns. Academic Press, London (1983)
4. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 2226–2231 (1996)
5. Silverman, B.: Density Estimation. Chapman and Hall, London (1986)
6. Bailey, T., Gatrell, A.: Interactive Spatial Data Analysis. Longman, Harlow (1995)
7. Bastin, L., Rollason, J., Hilton, A., Pillay, D., Corcoran, C., Elgy, J., Lambert, P., De, P., Worthington, T., Burrows, K.: Spatial aspects of MRSA epidemiology: a case study using stochastic simulation, kernel estimation and SaTScan. International Journal of Geographical Information Science 21, 811–836 (2007)
8. Mesev, V., Shirlow, P., Downs, J.: The geography of conflict and death in Belfast, Northern Ireland. Annals of the Association of American Geographers 99, 893–9035 (2009)
9. Ramp, D., Caldwell, J., Edwards, K., Warton, D., Croft, D.: Modelling of wildlife fatality hotspots along the snowy mountain highway in New South Wales, Australia. Biological Conservation 126, 474–490 (2005)
10. Woolford, D.G., Braun, W.J.: Convergent data sharpening for the identification and tracking of spatial temporal centers of lightning activity. Environmetrics 18, 461–479 (2007)

11. Brunsdon, C.: Estimating probability surfaces for geographical point data - an adaptive kernel algorithm. Computers & Geosciences 21, 877–894 (1995)
12. Worton, B.: Kernel methods for estimating the utilization distribution in home-range studies. Ecology 70, 164–168 (1989)
13. Hsieh1, J., Chen, S., Chuang, C., Chen, Y., Guo, Z., Fan, K.: Pedestrian segmentation using deformable triangulation and kernel density estimation. In: Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, pp. 3271–3274 (2009)
14. Righton, D., Mills, C.: Application of GIS to investigate the use of space in coral reef fish: a comparison of territorial behaviour in two Red Sea butterfly fishes. International Journal of Geographical Information Science 20, 215–232 (2006)
15. Hemson, G., Johnson, P., South, A., Kenward, R., Ripley, R., Macdonald, D.: Are kernels the mustard? Data from global positioning system (GPS) collars suggests problems for kernel home-range analyses with least-squares cross-validation. Journal of Animal Ecology 74, 455–463 (2005)
16. Mitchell, M.S., Powell, R.A.: Estimated home ranges can misrepresent habitat relationships on patchy landscapes. Ecological Modelling 216, 409–414 (2008)
17. Downs, J., Horner, M.: Effects of point pattern shape on home-range estimates. Journal of Wildlife Management 72, 1813–1818 (2008)
18. Borruso, G.: Network density estimation: Analysis of point patterns over a network. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005, part 3. LNCS, vol. 3482, pp. 126–132. Springer, Heidelberg (2005)
19. Okabe, A., Satoh, T., Sugihara, K.: A kernel density estimation method for networks, its computational method and a GIS-based tool. International Journal of Geographical Information Science 23, 7–32 (2009)
20. Wand, M., Jones, M.: Kernel Smoothing. Chapman and Hall, London (1995)
21. Hägerstrand, T.: What about people in regional science? Papers of the Regional Science Association 24, 7–21 (1970)
22. Miller, H.: A measurement theory for time geography. Geogr. Anal. 37, 17–45 (2005)
23. Miller, H., Bridwell, S.: A Field-Based Theory for Time Geography. Annals of the Association of American Geographers 99, 49–75 (2009)
24. Miller, H.: Necessary space-time conditions for human interaction. Environment and Planning B-Planning & Design 32, 381–401 (2005)
25. Kwan, M.: Interactive geovisualization of activity-travel patterns using three-dimensional geographical information systems: a methodological exploration with a large data set. Transp. Res. Pt. C-Emerg. Technol. 8, 185–203 (2000)
26. Neutens, T., Witlox, F., De Weghe, N., De Maeyer, P.: Space-time opportunities for multiple agents: A constraint-based approach. International Journal of Geographical Information Science 21, 1061–1076 (2007)
27. Winter, S.: Towards a probabilistic time geography. In: ACM GIS 2009, pp. 528–531 (2009)
28. Jiang, Z., Sugita, M., Kitahara, M., Takatsuki, S., Goto, T., Yoshida, Y.: Effects of habitat feature, antenna position, movement, and fix interval on GPS radio collar performance in Mount Fuji, central Japan. Ecol. Res. 23, 581–588 (2008)
29. Neutens, T., Witlox, F., Van de Weghe, N., De Maeyer, P.: Human interaction spaces under uncertainty. Transportation Research Record, 28–35 (2007)
30. Wentz, E.A., Campbell, A.F., Houston, R.: A comparison of two methods to create tracks of moving objects: linear weighted distance and constrained random walk. Int. J. Geogr. Inf. Sci. 17, 623–645 (2003)

Microtheories for Spatial Data Infrastructures - Accounting for Diversity of Local Conceptualizations at a Global Level

Stephanie Duce[1] and Krzysztof Janowicz[2]

[1] Department of Languages and Systems, University Juame I, Spain
[2] Department of Geography, The Pennsylvania State University, USA

Abstract. The categorization of our environment into feature types is an essential prerequisite for cartography, geographic information retrieval, routing applications, spatial decision support systems, and data sharing in general. However, there is no a priori conceptualization of the world and the creation of features and types is an act of cognition. Humans conceptualize their environment based on multiple criteria such as their cultural background, knowledge, motivation, and particularly by space and time. Sharing and making these conceptualizations explicit in a formal, unambiguous way is at the core of semantic interoperability. One way to cope with semantic heterogeneities is by standardization, i.e., by agreeing on a shared conceptualization. This bears the danger of losing local diversity. In contrast, this work proposes the use of microtheories for Spatial Data Infrastructures, such as INSPIRE, to account for the diversity of local conceptualizations while maintaining their semantic interoperability at a global level. We introduce a novel methodology to structure ontologies by spatial and temporal aspects, in our case administrative boundaries, which reflect variations in feature conceptualization. A local, bottom-up approach, based on non-standard inference, is used to compute global feature definitions which are neither too broad nor too specific. Using different conceptualizations of rivers and other geographic feature types, we demonstrate how the present approach can improve the INSPIRE data model and ease its adoption by European member states.

Keywords: Ontology, Geo-Semantics, Semantic Heterogeneity.

1 Introduction and Motivation

In 2007 the European Union launched the Infrastructure for Spatial Information in the European Community (INSPIRE) which aims at creating a Spatial Data Infrastructure (SDI) supporting cross-scale, cross-language, and cross-border interoperability and access to geodata[1]. This involves the development of spatial data themes, web services, agreements on data and service sharing, coordination and monitoring mechanisms, and especially also common metadata standards

[1] INSPIRE Directive http://inspire.jrc.ec.europa.eu/index.cfm

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 27–41, 2010.

and geographic feature (object) type catalogs. The European Union, however, is very heterogeneous in terms of ecosystems, climatic and physical conditions, cultures, languages, and administrative systems. This makes the definition of a shared conceptualization of geographic features a difficult task. If the guidelines set up by INSPIRE are too generic, i.e., do not sufficiently restrict possible interpretations [1], they will fail to establish interoperability or at least require manual, application specific, and error-prone adjustments. Overly specific guidelines could hinder implementation and reduce the usability of the data. In general, creating such a broad and multipurpose infrastructure to ensure overarching interoperability carries the danger that important nuances in the local and contextual terminology will be lost. For the INSPIRE initiative to be effective, efficient and successful, all parties should be free to define geographic feature types in a manner most suited to their unique environment and culture though still consistent at an all-encompassing upper level. This need introduces a struggle to create, integrate, and maintain conceptualizations at a local and European level.

The importance of local conceptualizations of geographic features has been widely acknowledged and discussed in the literature. Geographic features are susceptible to sorites vagueness and are characterized by vague boundaries [2, 3], vague adjective-based definitions [4, 5], meso-scale [6], and temporal dynamics [7]. This means that human perception, language, and social agreement play a strong role in our conceptualization of geographic features and can lead to semantic heterogeneities [8–11].

For instance, a forest can be a protected area, plantation, recreational area, agricultural area, habitat, and so forth. These different perspectives give rise to potential socio-economic conflicts but also hinder classification and retrieval. Lund [12], for instance, lists over 900 (often contradictory) definitions of forest. As forests do not stop at borders, a forest in Spain may be regarded as meadowland in France. Whether an area is categorized as forest or not may have legal and economic consequences as in the case of deforestation.

Given the indeterminacy of geographic features used for land cover classification and their increasing availability to the public, Comber and Fisher [10] argue that there is an urgent need for the semantics of data to be made explicit to users. An ontology for the geographic domain should reflect and capture multiple conceptualizations of geographic features [13].

The challenge of handling local [14], i.e., domain specific, conceptualizations at a global level is not new and has been a core topic in Artificial Intelligence (AI) research for 30 years [15, 16]. The key idea is to be consistent at the local level but allow contradicting conceptualizations within the global overall knowledge base. One promising approach to handle semantic heterogeneity is to structure knowledge in domain specific microtheories (also called contexts). This approach has been first implemented in the OpenCyC ontology which contains hundreds of thousands of terms and assertions. Each microtheory is designed as a coherent set of statements and can be thought of as a single ontology. Separate microtheories can hold information about the same concept but contain incompatible facts. For

instance, one microtheory may be strict about physical properties and laws of nature, while other microtheories may have weaker constraints to support *naïve physics* [17].

Usually microtheories are organized in subsumption hierarchies, i.e., facts specified in the super-microtheory must also hold in each of its sub-theories. Sibling-theories, however, may contain contradicting conceptualizations. Note that microtheories are not the only approach to ontology modularization [9, 18–20]. Kokla and Kavouras [21, 22] discussed the use of concept lattices to identify overlapping relationships and manage different geographic domain ontologies. While Guha et al. [23] revitalized the notion of context for the Semantic Web. Batemann et al. [19] discuss how to develop multi-perspectival ontologies of space using algebraic specifications and DOLCE as foundational ontology. The microtheories approach proposed here calculates a the Least Common Subsumer rather than using a concept lattice to identify commonalities and overlaps between different microtheories.

The main difference between our approach and previous work on microtheories is the use of alternative ordering principles. In previous work we proposed to introduce time and space as additional first class ordering principles for microtheories [24]. For instance, the definition of rivers differs markedly between southern European and northern European countries and hence microtheories specifying local conceptualizations may contradict. However, these microtheories have to be consistent with an EU-wide theory. As semantic hererogeneity is not a problem but a challenge, such an approach supports the diversity of different feature type conceptualizations across Europe, while creating and maintaining a consistent global ontology at a European scale to support interoperability. In this work, we discuss how microtheories can be used to define local conceptualizations and demonstrate how non-standard inference [25] and similarity reasoning can be employed to automatically infer an appropriate top-level as a common compromise. While our work is not restricted to INSPIRE or SDI, they will serve as running examples throughout the paper.

2 Structuring Microtheories by Administrative Containment

This section introduces the role of microthoeries and the methodology used to compute a top-level conceptualization from local knowledge.

The use of microtheories for knowledge representation and reasoning has numerous advantages [26] – the ability to support multiple conceptualizations for the same terminology and to provide structural relationships between these theories are the two most relevant benefits for the presented work. As each microtheory is considered an object in its own right and is only evaluated in a given context, two microtheories can hold conflicting facts without undermining the reasoning capacity of the entire knowledge base [26, 27]. In addition, microtheories provide modularity for ontologies [18–20]. This makes reasoning and querying more efficient as only relevant parts are used to answer a query [26].

Modularization also eases the updating of ontologies and allows their evolution without having to make widespread changes to the overall system. This is highly desirable as concepts in geospatial domains are regularly evolving as better understanding is achieved [28].

From the INSPIRE perspective, different conceptualizations of the same geographic feature may conflict with each other. Germany's conceptualization of river may state that it contains flowing water. However, in Spain, where rivers may by dry for most of the year, the definition of river cannot rely on the presence of flowing water. Most ontologies developed for the semantic (geospatial) Web [29], are strongly bound by the rules of logic and cannot cope with such conflicts. Therefore, in order to merge the definitions of rivers in Spain and Germany to create a Europe-wide conceptualization of rivers one of them would need to be changed to a definition that does not reflect the nature of the features in that country. This is undesirable and undermines the success of the INSPIRE initiative. Rivers and forests are by no means the only examples - in fact most terminologies require a spatially bounded context for their interpretation.

2.1 Structuring Microtheories

So far, microtheories have only been structured by establishing hierarchical relationships between them, i.e, by generalization. Other potential ordering principles such as space, time, or cultural background have received nearly no attention in the Semantic Web community. While their importance has been recognized recently, existing work reduces space and time to simple latitude-longitude pairs and time stamps. Tobler's First Law of Geography states that 'Everything is related to everything else, but near things are more related than distant things'. Climatic, geographic and geological factors, all of which adhere to the above law, govern the character of geographic features and hence influence their categorization. Besides their role in the gradual change of the environment, space and time are the most fundamental ordering relations used in human cognition and language – spatial metaphors are just one prominent example [30].

More formally, the hierarchy of microtheories is created using a generalization relationship between microtheories called *genlMt* in OpenCyC and *specializes* by McCarthy and Buvac [31]. If $ist(mt, p)$ is the *is true in* relation between a microtheory mt and a predicate p, then $genlMt$ is the anti-symmetric, reflexive, and transitive, binary predicate by which the theory hierarchy is constructed by adding axioms of the form

$$mt_0 : \ \forall p \ ist(mt_g, p) \wedge genlMt(mt_g, mt_s) \longrightarrow ist(mt_s, p)$$

to the topmost theory mt_0; where mt_g is the more general and mt_s the more specific theory[2]. Figure 1a depicts the relation between an overall geographic microtheory and a more specific version for volunteered geographic information. The second may introduce new vocabulary for navigation, such as landmarks,

[2] Note that we do not allow cycles; see [23] for details.

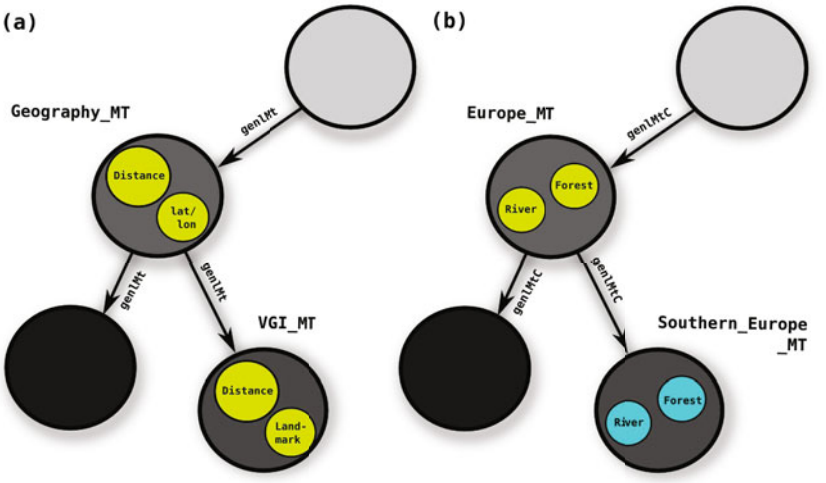


Fig. 1. Structuring microtheories by (a) generalization (*genlMt*) and by (b) generalization and (spatial) containment (*genlMtC*); see also [24].

instead of relying on latitude and longitude only and may redefine the notion of distance.

To structure microtheories by spatial (or administrative) containment, we introduce the *genlMtC* relation which extends *genlMt* as follows:

$$mt_0 : \forall p\ ist(mt_g, p) \wedge genlMtC(mt_g, mt_s) \longrightarrow genlMt(mt_g, mt_s) \wedge \circledcirc(mt_g, mt_s)$$

Consequently, $genlMtC(mt_g, mt_s)$ holds if mt_s is a sub-theory of mt_g and all footprints of individuals of geographic feature types specified in mt_s are (spatially or administratively) contained in mt_g[3]. Examples, depicted in figure 1, include the river and forest case discussed previously.

In the following, we will use these relationships as meta-theory for local ontologies represented using description logics. Hence, a more detailed specification of the containment relation is left for further work.

2.2 Spatial versus Administrative Containment

Dividing Europe into appropriately structured microtheories using spatial containment is difficult. A division by geographic factors such as climate and geology may lead to scale problems and especially, to administrative challenges as one country could fall into more than one theory and multiple countries could belong to the same microtheory. A more fine grained solution would be to decide to which theory each feature type belongs. However, this would again be impractical from an administrative perspective.

[3] The second part is denoted by the $\circledcirc$-predicate and requires a spatial footprint for the individuals as well as for the spatial scope of the theory. A formal semantics for $\circledcirc$ including RCC is left for further work.

This paper describes the possible structuring of microtheories based on administrative boundaries. This method takes geographic and climatic factors into account (to some extent) and offers intuitive divides from a political perspective. Using this method each EU member state would define its own microtheories, best reflecting the conceptualization of geographic features in its country. This would overcome some of the administrative difficulties and align well with present data models which are usually created on the national level.

Nevertheless, administrative structuring is not ideal as the territories of countries are large and diverse themselves. Also, a country may possess outside territory where geographic features may be very different (e.g., the UK and Gibralta). To overcome these issues, autonomous or independent regions could make their own microtheory where necessary. A nation-wide microtheory could then be generalized from the internal regions. We cannot offer a definitive solution here as multiple situations may require different choices. As INSPIRE acts as a running example to illustrate our theoretical approach, we use administrative containment in the following.

2.3 Methodology

Features such as rivers, forests, and estuaries demonstrate the benefits of the microtheory-approach as their conceptualization is strongly based on factors that vary in space (rainfall, geology, topography...). They are of great importance from economic, social and environmental perspectives and involve various stakeholders. An effective SDI, based on well defined and semantically interoperable feature definitions, is imperative to understand, study and successfully manage these features.

Several steps are required to demonstrate the use of a bottom-up approach to compute an appropriate[4] global definition as a compromise between local conceptualizations.

1. Natural language definitions of geographic feature types have to be selected from the literature. These definitions should reflect the local (i.e., country specific) viewpoints. In our case, we present definitions for the feature *River*. Spain and Germany were chosen for treatment as their rivers represent different ends of the spectrum of contrasting river conditions across Europe. These natural language definitions are expanded into concept maps and related to other features in the domain.
2. To support non-standard inference and similarity reasoning these definitions are formalized using the Web Ontology Language (OWL) and the Protégé editor. Note that some parts of the informal concept map definitions cannot be adequately represented in OWL.
3. To generate an appropriate top-level for the global ontology, the Least Common Subsumer (LCS) [25] will be computed as it fulfils the requirements of *appropriateness* described above. The computation of the LCS between

[4] Appropriate is defined here as a conceptualization that is neither too broad nor too specific in the number and type of geographic features that are covered.

DL-based concepts requires a trade off between the expressivity of the conceptualizations and the reasoning capabilities of the methodologies and tools used. Similarity reasoning and computing the LCS can only be performed on a subset of OWL. Hence, further reductions to the concept maps are required. In many cases, these restrictions are caused by the tools selected and can be resolved in the near future with new implementations. One typical example, namely the problem of handling logical disjunction in case of the LCS will be discussed in the formalization section.

4. Finally, after computing the LCS – which in our running example serves as the EU wide definition of *River* – we use subsumption and similarity reasoning to evaluate our results, i.e., to check whether the LCS provides a more appropriate top-level conceptualization than the existing INSPIRE definition(s).

Similarity reasoning, using the SIM-DL reasoner, is employed to test how well the definitions capture the domain and reflect human conceptualizations. SIM-DL is an asymmetric, context-aware similarity measurement theory used for information retrieval. It compares a DL search concept with one or more target concepts, by measuring the degree of overlap between their definitions. See [32, 33] for details on SIM-DL and similarity estimations.

The least common subsumer (computed in step 3) is defined as follows:

Definition 1. *Given a description logic $\mathcal{L}$, and a set of concepts $C_1, ..., C_n$, a particular concept D is the least common subsumer with respect to $C_1, ..., C_n$ iff it satisfies the following conditions:*

(a) $C_i \sqsubseteq D$ for all $C_{1,...,n}$
(b) All concepts D' satisfying $C_i \sqsubseteq D'$ (for all $C_{1,...,n}$) also satisfy $D \sqsubseteq D'$, i.e., D is the least $\mathcal{L}$ concept satisfying (a) and unique.

3 Application

This section demonstrates the application of the above methodology. We introduce local microtheories for rivers in Spain and Germany. These are related via *genlMtC* to the EU microtheory which was computed using the LCS i.e., the least common subsumer of the conceptualizations provided by the local microtheories. While the full study also includes forests and estuaries in the theories we focus on the river example due to lack of space.

3.1 Natural Language Definitions

The traditional northern European perspective of a river is a continuously flowing body of water which may also be navigable [34]. This view is reflected in the INSPIRE context. We especially refer to the *INSPIRE Feature Concept Dictionary*[5], the *INSPIRE Consolidated UML Model*[6], the *EuroRegionalMap*

[5] https://inspire-registry.jrc.ec.europa.eu/registers/FCD
[6] http://inspire-twg.jrc.ec.europa.eu/inspire-model/; Generatied 24 August 2009 v3, Revision 873.

Specification and Data Catalogues, as well as the *Water Framework Directive (WFD)*. In these classifications, *Watercourse* is defined as '[a] natural or man–made flowing watercourse or stream'[7], while *WFD River* is defined as '[a] body of inland water flowing for the most part on the surface of the land but which may flow underground for part of its course'[8]. This definition seems broad, however, its requirement of flowing water may be too specific to encompass rivers in the Mediterranean climes of southern Europe – especially taking the effects of global warming into account. For example, rivers in southern Spain are highly ephemeral and may only contain water during flood events. In these regions, the conceptualization of rivers may include channels or depressions through which water flows, even if they are dry [35].

Rivers are highly complex ecosystems and commercialized anthropogenic entities. Hence, the definitions presented in this work do not claim to encompass all their elements. They show how diverse elements can be used to better define local conceptualizations without undermining global interoperability. The ecosystem functions and anthropogenic services performed by rivers are considered in this work to be (thematic) *roles*, similar to the notion of affordances used for modeling by Kuhn [36] and others in GIScience.

These properties of rivers may to some extent transcend the spatio-temporal vagueness and variability which hamper the use of mereotopology in defining rivers and are likely to represent commonalities and distinctions in different local conceptualizations. For example, rivers, wherever they are, play the role of transporting water. In Germany a river can also play the role of providing transport to humans and goods (as their constant flow of water makes them navigable). However, in Spain the frequent lack of water means rivers are not perceived as navigable. In the following, rivers in Spain and Germany are defined in natural language terms. These definitions were derived from multiple sources to ensure they are not biased by a particular point of view.

A Spanish river is a channel, with a bed and more or less defined banks, which transects a river basin at a low point in the topography. It drains water which falls as precipitation on the river basin. It has a flow regime which refers to the average presence or absence of water within the channel throughout a year. It may participate in flood events and provides the ecological service of protecting against these events. Spanish rivers also participate in droughts and can provide terrestrial or aquatic habitat, terrestrial or aquatic recreational areas and play the role of supplying water.

A German river is a channel, with a river bed and river banks which contains flowing water and transects a river basin with another waterbody as its destination. It represents the above ground expression of the groundwater table and also drains water, from precipitation or snow melt, in the river basin. It may participate in flood events and provides the ecological service of protecting against

[7] http://inspire-registry.jrc.ec.europa.eu/registers/FCD/items/105 as of 05-Dec-08
[8] http://inspire-registry.jrc.ec.europa.eu/registers/FCD/items/421 as of 19-Jan-10

these events. German rivers provide aquatic habitat and aquatic recreational areas and play the role of supplying water and transportation.

3.2 Conceptual Modeling

According to the presented methodology, the natural language definitions are encoded as semi-formal concept maps and aligned to the top-level classes, *Physical Endurant*, *Perdurant*, *Role* and *Quality*, proposed by the DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) top-level ontology [37]. The concept maps are depicted in the figures 2 and 3 with the main differences marked red[9]. They show that German rivers contain flowing water and have river banks as their proper parts. However, for Spanish rivers these relationships are optional (indicated by dotted lines) to reflect the temporal variability and vagueness of these properties. Furthermore, German rivers are defined as having waterbodies as their destinations which is not required for Spanish rivers as they may simply peter out. German and Spanish rivers were defined as having precipitation and ground water as sources of water with Germany having snow melt as an additional source. Flood events (and drought events in Spain) as well as erosion are of particular management importance and thus were included in the conceptual models.

The definitions deliberately avoid reference to rivers being artificial or natural as these terms are vague and can cause confusion. For example, natural rivers can have artificial components (e.g. bank stabilization measures) or an artificial flow regime (e.g., due to the presence of a dam). While these characteristics may help distinguish between some feature types (e.g., canal and river), they do not provide identity to rivers. To support grounding by observations, we also include properties such as water depth to be linked to the currently developed measurement ontologies [38].

3.3 Formalization

To use the Semantic Web infrastructure and reasoners the conceptual models are represented in OWL. The Protégé versions 4 and 3.3.1 were used as ontology editors as they provides plug-ins to the SIM-DL reasoner which supports, subsumption and similarity reasoning as well as the computation of the LCS [32, 33][10]. Several simplifications and ontological commitments are necessary to represent the conceptual models of rivers. For instance, due to the open world assumption, optional relations are not specified in the river definitions.

Most importantly, the LCS is only meaningful for description logics without disjunction as the LCS would simply be the disjunction of compared concepts. There are two solutions to this problem. First, to reduce the expressivity of the

[9] The presented conceptual models are simplified for reasons of readability; the original and more detailed versions can be downloaded at http://www.personal.psu.edu/kuj13/GIScience2010MT.zip.

[10] Note that the current version only supports a subset of OWL-DL and the computation of the LCS is even further restricted; download at http://sim-dl.sf.net/.

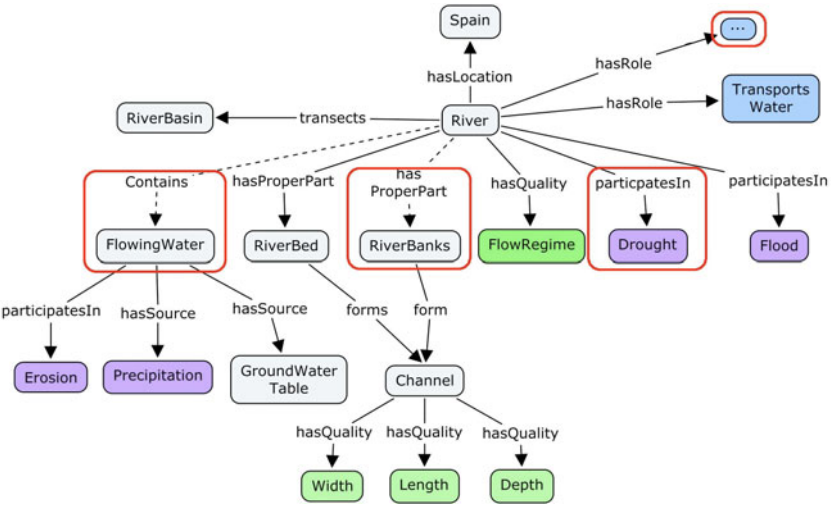


Fig. 2. Simplified conceptual model showing the relations between entities defining a *SpanishRiver*. The entities are divided roughly into the DOLCE top-level classes: physical endurant (white), perdurant (purple), role (blue) and quality (green). Elements of difference to the German river definition are marked in red.

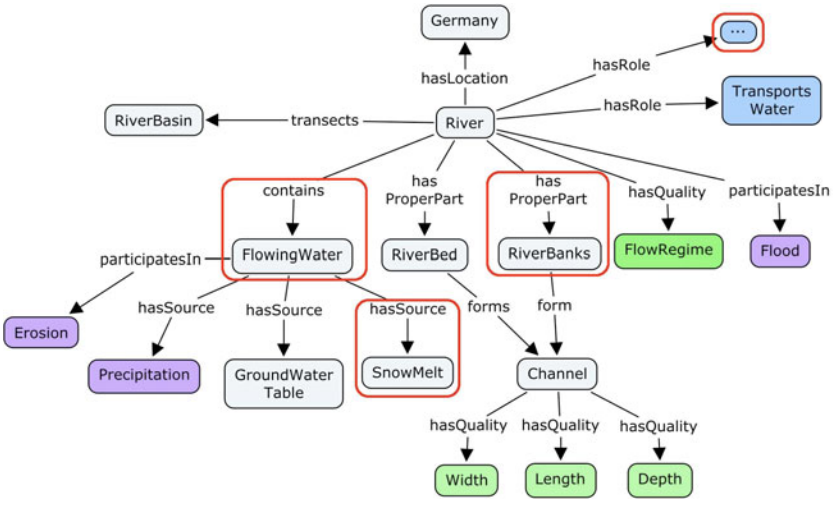


Fig. 3. Simplified conceptual model showing the relations between entities defining a *GermanRiver*. Elements of difference to the Spanish river definition are marked in red. This model has been simplified for presentation.

language used and hence approximate the conceptualizations, e.g., by vivification [39]. Second, to compute a *good* instead of the *least* common subsumer; see [40] for details.

The German and Spanish river classes are not modeled as disjoint because, given their broad scale, a single river can have multiple conceptualizations, multiple links and may fall within one or more member states. Instances may be attributed to one, or more than one, of the microtheories. Figure 4 shows a fragment of the ontology. Note that we have combined the definitions of German, Spanish, and EU rivers in a single ontology to perform reasoning in Protégé. In fact, they are in separate microtheories and hence are all named *River*. Difficulties arising from semantic heterogeneities are captured by our approach as all definitions have a spatial (or in our case administrative) context.

3.4 Computing the Top-Level

The spirit of modeling on the Semantic Web has often been confused. In contrast to specifying multiple taxonomies by hand, the driving idea is to let the reasoner do the *untangling*, i.e., reclassify a developed ontology, discover, and add implicit subsumption relations. In this spirit, but using the reverse direction, we propose to specify the local and member state specific conceptualizations and let the reasoner compute the common top-level. Consequently, we do not use subsumption reasoning but compute the least common subsumer as most specific top-level concept for each feature type. A similar approach was also proposed and implemented into SIM-DL in previous work [41]. Computing *EuropeanRiver* as the last common subsumer of the German and Spanish river definitions yields:

$$EuropeanRiver \equiv \exists.transects(RiverBasin) \sqcap \exists.hasPart(RiverBed) \sqcap ... \sqcap \exists.hasLocation(MemberState) \sqcap \exists.hasQuality(FlowRegime) \sqcap [\texttt{vivification}]^{11}$$

As the Spanish definition lacks the *contains FlowingWater* and *hasPart RiverBanks* restrictions, these were excluded from the *EuropeanRiver* definition. The common filler between the two definitions for the *hasLocation* property, *MemberState*, was used. Thus, based on the above methodology, a European river transects a river basin and has a river bed, flow regime, is located in an EU member state, and performs a suite of roles. Reclassification of the ontology showed that the *SpanishRiver* and *GermanRiver* are subsumed by *EuropeanRiver*, while the INSPIRE definition of river excludes the *SpanishRiver* (see figure 4). Consequently, the INSPIRE definition is too specific even when just two local definitions are compared. Adding more definitions from other member states is likely to further broaden the EU definition. While restricting rivers to flowing watercourses is too exclusive, in other respects the INSPIRE definition is too generic and could be more specific (and hence improving semantic interoperability). For instance, the river definition could include a relation to river basins; especially as *RiverBasin*[12] is already listed in the INSPIRE Feature Concept Dictionary.

[11] A richer approximation of the LCS could be determined by vivifivation or by computing the *good common subsumer*; both require manual interaction.

[12] http://inspire-registry.jrc.ec.europa.eu/registers/FCD/items/409 as of 19-Jan-10

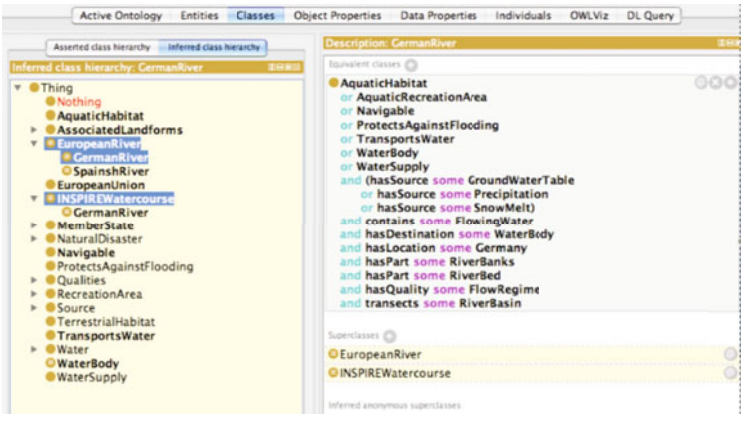


Fig. 4. Screenshot showing the inferred class hierarchy and the restrictions used to define *GermanRiver*. The reasoner inferred that *GermanRiver* and *SpanishRiver* are both kinds of *EuropeanRiver* (as this is the LCS). The INSPIRE *Watercourse* (and *River*) can only act as superclass for the *GermanRiver* and excludes the *SpanishRiver*.

3.5 Similarity Results

Finally, *EuropeanRiver* was compared to the other definitions using the SIM-DL similarity server running the *maximum* & *asymmetry* modes [32]. As expected, from the perspective of the European river definition, the similarity to the German and Spanish definitions is 1.0. This means that users searching for *EuropeanRiver* will be satisfied retrieving both kinds of river (e.g., using a semantics-enabled interface for Web gazetteers [32]). This is not surprising as the EU wide definition is the super concept of both. The comparison to the INSPIRE definition results in a low similarity (0.11). This is to be expected as SIM-DL measures the conceptual overlap and the INSPIRE definition does not contain several of the statements made for the EU wide definition. Note, however, that adding more member states would broaden the European river definition and make it more similar to the INSPIRE version.

4 Conclusions and Further Work

In this work we have discussed the importance of local conceptualizations of geographic space. Different communities have developed their own understanding and terminology for good reasons [14]. By introducing microtheories and structuring them by spatial or administrative containment, we have shown how local and potentially contradictory conceptualizations can be reconciled in a common knowledge base. Next, we have presented a methodology to compute the top-level of such knowledge bases instead of standardizing common feature types manually – which may exclude local conceptualizations. We have tested our approach by specifying local river definitions and computing the least common subsumer

as a common EU wide definition. Our results show that the definitions proposed by INSPIRE and the Water Framework Directive are too specific in some respects while lacking other relations, e.g., to river basins. The presented EU wide definition could contain more details and hence be a better approximation of a common compromise. As argued above, this could be done semi-automatically by computing the good common subsumer or using vivification. Both approaches are promising and will be investigated in future work. Our full study also takes forests and estuaries from multiple member states into account; the concept maps and ontologies are available online. While incorporating ontologies and Semantic Web reasoners into SDIs has been difficult so far, recent work on Semantic Enablement for Spatial Data Infrastructures may ease their integration [42]. A first reference implementation of a Web Reasoning Service (WRS) for similarity reasoning is available online at the 52°North semantics community[13].

The structuring of microtheories by spatial and temporal relations presented here gives initial insights into the role of space and time for ontology modularization [24]. However, the approach is still at an early stage and requires an improved and rigid, formal underpinning. The work also points to limitations in the ability of existing Semantic Web representation languages and reasoners to adequately deal with the expressive conceptualizations necessary to effectively define vague, dynamic and highly interlinked features in geographic space. Future work should especially also focus on machine learning approaches to derive the top-level conceptualizations and should be compared to the results of the deductive approach taken in this work.

References

1. Kuhn, W.: Semantic engineering. In: Navratil, G. (ed.) Research Trends in Geographic Information Science, pp. 63–74. Springer, Heidelberg (2009)
2. Fisher, P.: Sorites paradox and vague geographies. Fuzzy Sets and Systems 113, 7–18 (2000)
3. Smith, B., Mark, D.M.: Do mountains exist? towards an ontology of landforms. Environment and Planning B: Planning and Design 20(2), 411–427 (2003)
4. Mark, D.M.: Toward a theoretical framework for geographic entity types. In: Campari, I., Frank, A.U. (eds.) COSIT 1993. LNCS, vol. 716, pp. 270–283. Springer, Heidelberg (1993)
5. Bennett, B., Mallenby, D., Third, A.: An ontology for grounding vague geographic terms. In: Formal Ontology in Information Systems - Proceedings of the Fifth International Conference (FOIS 2008), vol. 183, pp. 280–293. IOS Press, Amsterdam (2008)
6. Smith, B., Mark, D.M.: Ontology and geographic kinds. In: International Symposium on Spatial Data Handling, Vancouver, Canada, pp. 308–320 (1998)
7. Frank, A.: A linguistically justified proposal for a spatiotemporal ontology. In: Kuhn, W., Worboys, M.F., Timpf, S. (eds.) COSIT 2003. LNCS, vol. 2825. Springer, Heidelberg (2003)
8. Egenhofer, M., Mark, D.M.: Naive geography. In: Kuhn, W., Frank, A.U. (eds.) COSIT 1995. LNCS, vol. 988, pp. 1–15. Springer, Heidelberg (1995)

[13] http://www.52north.org/semantics

9. Bishr, Y.: Overcoming the semantic and other barriers to gis interoperability. International Journal of Geographical Information Science 12(4), 299–314 (1998)
10. Comber, A., Fisher, P.: What is land cover? Environment and Planning B: Planning and Design 32, 199–209 (2005)
11. Kuhn, W.: Geospatial semantics: Why, of what and how? Journal of Data Semantics III, 1–24 (2005)
12. Lund, H.G.: Definitions of forest, deforestation, afforestation, and reforestation. Technical report, Forest Information Services (2009), http://home.comcast.net/~gyde/DEFpaper.htm
13. Janowicz, K., Maue, P., Wilkes, M., Schade, S., Scherer, F., Braun, M., Dupke, S., Kuhn, W.: Similarity as a quality indicator in ontology engineering. In: Eschenbach, C., Grueninger, M. (eds.) 5th International Conference on Formal Ontology in Information Systems (FOIS 2008), pp. 92–105. IOS Press, Amsterdam (2008)
14. Uschold, M.: Creating, integrating and maintaining local and global ontologies. In: Horn, W. (ed.) Proceedings of 14th European Conference on Artificial Intelligence (ECAI 2000). IOS Press, Amsterdam (2000)
15. McCarthy, J.: Generality in artificial intelligence. Communications of the ACM 30, 1030–1035 (1987)
16. Wachsmuth, I.: The concept of intelligence in ai. Prerational Intelligence – Adaptive Behavior and Intelligent Systems without Symbols and Logic 1, 43–55 (2000)
17. Smith, B., Casati, R.: Naive physics: An essay in ontology. Philosophical Psychology 7/2, 225–244 (1994)
18. Grau, B., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In: 20th International Joint Conference on Artificial Intelligence, pp. 183–196 (2007)
19. Bateman, J., Borgo, S., Luettich, K., Masolo, C., Mossakowski, T.: Ontological modularity and spatial diversity. Spatial Cognition and Computation 7, 97–128 (2007)
20. Hois, J., Bhatt, M., Kutz, O.: Modular ontologies for architectural design. In: Ferrario, R., Oltramari, A. (eds.) Formal Ontologies Meet Industry, pp. 66–78. IOS Press, Amsterdam (2009)
21. Kokla, M., Kavouras, M.: Fusion of top-level and geographic domain ontologies based on context formation and complementarity. International Journal of Geographical Information Science 15, 679–687 (2001)
22. Kavouras, M., Kokla, M.: A method for the formalization and integration of geographic categorizations. International Journal of Geographical Information Science 16, 439–453 (2002)
23. Guha, R., Mccool, R., Fikes, R.: Contexts for the semantic web. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 32–46. Springer, Heidelberg (2004)
24. Janowicz, K.: The role of place for the spatial referencing of heritage data. In: The Cultural Heritage of Historic European Cities and Public Participatory GIS Workshop, The University of York, UK, September 17-18 (2009)
25. Küsters, R.: Non-Standard Inferences in Description Logics. In: Küsters, R. (ed.) Non-Standard Inferences in Description Logics. LNCS (LNAI), vol. 2100, p. 33. Springer, Heidelberg (2001)
26. Cycorp: Contexts in cyc (2002), http://www.cyc.com/cycdoc/course/contexts-basic-module.html
27. Hovy, E.: Comparing Sets of Semantic Relations in Ontologies. In: The Semantics of Relationships: An Interdisciplinary Perspective. Kluwer Publishers, Dordrecht (2002)

28. Brodaric, B., Gahegan, M.: Distinguishing instances and evidence of geographical concepts for geospatial database design. In: Egenhofer, M.J., Mark, D.M. (eds.) GIScience 2002. LNCS, vol. 2478, pp. 22–37. Springer, Heidelberg (2002)
29. Egenhofer, M.: Toward the semantic geospatial web. In: GIS 2002: Proceedings of the 10th ACM international symposium on Advances in geographic information systems, pp. 1–4. ACM, New York (2002)
30. Lakoff, G., Johnson, M.: Metaphors We Live By. University Of Chicago Press, Chicago (1980)
31. McCarthy, J., Buvac, S.: Formalizing context (expanded notes) (1996)
32. Janowicz, K., Kessler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., Baeumer, B.: Algorithm, implementation and application of the sim-dl similarity server. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 128–145. Springer, Heidelberg (2007)
33. Janowicz, K., Wilkes, M.: SIM-DL_A: A novel semantic similarity measure for description logics reducing inter-concept to inter-instance similarity. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 353–367. Springer, Heidelberg (2009)
34. Taylor, M., Stokes, R.: Up the creek: What is wrong with the definition of a river in new south wales? Environment and Planning Law Journal 22(3), 193–211 (2005)
35. Taylor, M.P., Stokes, R.: When is a river not a river? consideration of the legal definition of a river for geomorphologists practising in new south wales, australia. Australian Geographer 36(2), 183–200 (2005)
36. Kuhn, W.: Ontologies in support of activities in geographical space. International Journal of Geographical Information Science 15(7), 613–631 (2001)
37. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltrami, A.: Ontology library deliverable d18. Technical report, ISTC-CNR (2003)
38. Kuhn, W.: A functional ontology of observation and measurement. In: Janowicz, K., Raubal, M., Levashkin, S. (eds.) GeoS 2009. LNCS, vol. 5892, pp. 26–43. Springer, Heidelberg (2009)
39. Cohen, W., Borgida, A., Hirsh, H.: Computing least common subsumers in description logics. In: Proceedings of the 10th National Conference on Artificial Intelligence, pp. 754–760. MIT Press, Cambridge (1992)
40. Baader, F., Sertkaya, B., Turhan, A.Y.: Computing the least common subsumer w.r.t. a background terminology. Journal of Applied Logic 5(3), 392–420 (2007)
41. Janowicz, K., Wilkes, M., Lutz, M.: Similarity-based information retrieval and its role within spatial data infrastructures. In: Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 151–167. Springer, Heidelberg (2008)
42. Janowicz, K., Schade, S., Bröring, A., Keßler, C., Maue, P., Stasch, C.: Semantic enablement for spatial data infrastructures. Transactions in GIS 14(2), 111–129 (2010)

The Family of Conceptual Neighborhood Graphs for Region-Region Relations

Max J. Egenhofer

National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
Department of Computer Science
University of Maine
Boardman Hall, Orono, ME 04469-5711, USA
max@spatial.maine.edu

Abstract. This paper revisits conceptual neighborhood graphs for the topological relations between two regions, in order to bridge from the A-B-C neighborhoods defined for interval relations in R^1 to region relations in R^2 and on the sphere S^2. A categorization of deformation types—built from *same* and *different* positions, orientations, sizes, and shapes—gives rise to four different neighborhood graphs. They include transitions that are constrained by the regions' geometry, yielding some directed, not undirected neighborhood graphs. Two of the four neighborhood graphs correspond to type B and C. The lattice of conceptual neighborhood graphs captures the relationships among the graphs, showing completeness under union and intersection.

Keywords: Spatial reasoning, topological relations, 9-intersection, conceptual neighborhood graphs.

1 Introduction

The increasing interest in *qualitative* spatial relations to describe high-level spatial information fosters the development of new analytical methods that make logical inferences on such relations. Qualitative spatial relations abstract away the myriad of quantitative details that one may observe between spatial objects, focusing on properties that are deemed most important to capture how such spatial objects are related. Representing continuous properties by discrete systems of symbols and providing calculi for reasoning with spatial entities is the essence of qualitative reasoning [7]. Consistently defined sets of qualitative spatial relations form a foundation for spatial constraints in spatial query languages [11], support the definition of the semantics of natural-language spatial predicates [23], and enable computational methods to reason about such spatial relations without the use of a graphical depiction to visually deduce the potential configurations [9]. Most research has focused on the *definition* of qualitative relations. Existing methods for modeling spatial relations have been comprehensively compiled in several survey articles [8, 20]. While early approaches [19, 26]

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 42–55, 2010.

addressed spatial relations in an integrated fashion, the use of tailored methods for different types—topological, direction, and metric—has prevailed during the last decade. Current formalizations for *topological* relations fall primarily into two major categories: (1) those based on connection [24] and (2) those based on intersection [15,16].

Since qualitative relations *per se* are essentially on a nominal scale, they only provide limited opportunities for making comparisons or analyses beyond an exact matching. To overcome such limitations, two methods have shown particular support for qualitative reasoning: (1) the *composition* of relations to infer a third relation from the two relations that hold over a common object and (2) each relation's *conceptual neighbors* to determine the most similar relations. All three facets—formalization, conceptual neighborhoods, and compositions—have been developed for topological relations between simple regions in R^2 [10,13,14,24,25] as well as for intervals in R^1 [2,18], and a plethora of similar components exist for topological relations of directed lines [22], complexly structured objects [29], convex regions [5], regions with indeterminate boundaries [4,6], broad lines [27], and line segments [28].

Conceptual neighborhood graphs have been critical for spatial similarity querying, when the specified relation needs to be relaxed because no data match the particular relation [3,9]. The graphs have also been used in cognitive similarity assessments [21] and as a foundation for the formalization of natural-language spatial predicates [23].

This paper analyzes the conceptual neighborhood graphs for topological relations between two regions in S^2 [12] (Fig. 1).

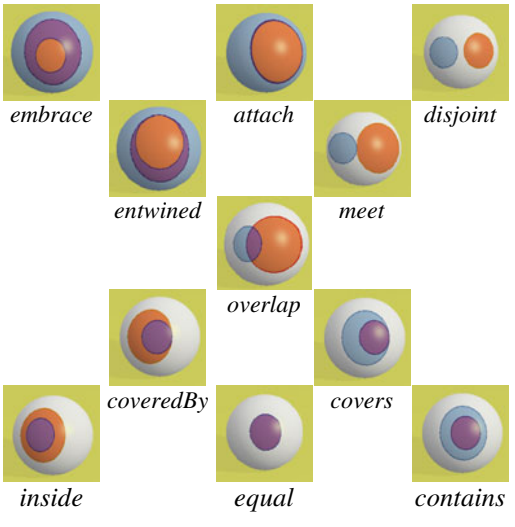


Fig. 1. Examples of the eleven topological relations between two regions in S^2 [12]

This set was chosen because it encompasses the eight popular topological relations between two regions in R^2 [14,24], but also includes another three relations that are exclusively spherical. These additional relations are expected to provide new insights about the interpretation of patterns in the neighborhood graphs. The eleven spherical

region-relations' complete graph with 55 edges forms the complete set of candidates for conceptual neighbors, but only a small subset is typically considered.

The remainder of the paper is structured as follows: Section 2 introduces a categorization of deformations to spatial regions. Section 3 derives for five simple deformations the conceptual neighbors of the region-region relations in S^2. Section 4 compares them with the A-B-C neighborhoods, and Section 5 constructs a lattice framework to establish the relationships among the family of conceptual neighborhood graphs. The paper closes with conclusions in Section 6.

2 Types of Deformations

A critical aspect in order to determine whether two qualitative relations are *conceptual neighbors* or not is the choice of *deformations* that apply to an object, since different types of deformations may lead to different *most similar* relations. Two relations are considered to be *conceptual neighbors* if the relations can be transformed into one another by continuously deforming the related objects in a topological sense [18]. For interval relations in R^1 three neighborhoods have been defined [18]: *A-neighbors* result from fixing three of the four endpoints of two intervals, while changing the fourth point; *B-neighbors* are the outcome of fixing the intervals' durations and moving these time-constrained intervals; and *C-neighbors* originate from varying an interval's duration while "leaving the 'temporal location' (reflected for instance by the midpoint)" fixed. Although these definitions do not scale up immediately from intervals in R^1 to regions in R^2, similar distinctions of neighborhood graphs have been made for region-region relations albeit with a different formalization [13].

In an effort to unify these approaches, we introduce a categorization of deformations upon objects as a rationale for distinguishing the different types of conceptual neighborhood graphs. The categorization is based on four object properties: (1) An object's *position* changes by moving the object; (2) its *orientation* is changed by rotating the object around a reference point; (3) the object's *size* is changed by increasing or decreasing it; and (4) its *shape* is changed through topological transformations that do not necessarily affect position, orientation, or size. Each of these four properties may change independently so that the remaining three properties remain unchanged. For instance, moving an object will not change its size, orientation, or shape. None of the deformations, however, will change the topological structure of an object. For instance, a 2-dimensional object will remain 2-dimensional and it will retain the number of holes and the number of separations.

These four properties are generic enough to apply to objects of different dimensions as well to embeddings of different dimensions, although for low-dimensional objects some of the deformations may have no effect. For example, the change in size by increasing the object applies to a 1-dimensional interval in R^1 as well as to a 2-dimensional region in R^2, while it has no effect on a 0-dimensional point. If an n-dimensional object is embedded in a higher-dimensional space (i.e., the object's codimension is great than 0), then some deformations may apply to multiple object parts. For instance, for a region embedded in R^3 the change in shape applies to the region's boundary as well as to the regions' interior. Likewise, a line embedded in R^2

may change its shape (while the same deformation to a line in R^1 would be immaterial). Since the focus is on relations between 2-dimensional regions on the sphere S^2, each property applies independently only to the object as a whole.

For each of these four properties we consider a set of coarse values, distinguishing whether a property remains the *same* or whether it is *different* after a deformation. The values *same* and *different* are mutually exclusive (i.e., two things cannot be simultaneously the *same* and *different*) and provide a complete coverage (i.e., there is no third possible state). These attributes apply to values; therefore, the common terminology found in *topology* [1]—*invariant* and *variant*—are reserved to properties. These distinctions—four categories with two values (*same*/*different*)—yield 2^4=16 different types of deformations (Fig. 2). Their 4-tuples of *same*/*different* specifications are jointly exhaustive and pairwise disjoint (much like the popular sets of basic topological relations). Further refinements are possible by considering additional criteria to distinguish types of difference. For instance, with respect to size one could refine *different* with *bigger* and *smaller*, yielding twice as many deformations with respect to size. Among the sixteen deformation categories resulting from *same* and *different*, four—move, rotate, isotropic scale, and anisotropic size-neutral deform—depend on a single occurrence of *different*, relying on a change in only one of the four properties; one is the neutral element (with all four values *same*); and eleven are combinations of differences in two, three, or all four properties.

	Deformation	Position	Orientation	Size	Shape
D0:	neutral	same	Same	same	same
D1:	move	different	same	same	same
D2:	rotate	same	different	same	same
D3:	isotropic scale	same	same	different	same
D4:	anisotropic size-neutral deform	same	same	same	different
D5:	move + rotate	different	different	same	same
D6:	move + isotropic scale	different	same	different	same
D7:	move + anisotropic size-neutral deform	different	same	same	different
D8:	rotate + isotropic scale	same	different	different	same
D9:	rotate + anisotropic size-neutral deform	same	different	same	different
D10:	anisotropic scale	same	same	different	different
D11:	isotropic scale	different	different	different	same
D12:	move + rotate + anisotropic size-neutral deform	different	different	same	different
D13:	move + anisotropic scale	different	same	different	different
D14:	rotate + anisotropic scale	same	different	different	different
D15:	move + rotate + anisotropic scale	different	different	different	different

Fig. 2. The sixteen deformation categories based on *same*/*different* position, orientation, size, and shape

Each of these 16 deformation types may be applied to one of the two related objects, providing a rationale for conceptual neighborhoods that is not limited to specific

types of spatial objects. If the same amount of change is applied to *both* objects *simultaneously*, their positions, orientations, sizes, and shapes may be different, but their topological relations will remain unchanged. Although the neighborhoods are only derived informally without formal proofs, they are exhaustive. As neighbor we consider any immediate relation obtained from a deformation (i.e., the relation obtained without a need to go through another relation). In this sense, deformation refers to *smallest change possible*.

3 Neighbors from Simple Deformations

Among the 15 meaningful deformations, we select the five that can occur on their own and analyze for each simple deformation the conceptual neighborhoods that they generate (Sections 3.1-3.3). Section 3.4 analyzes the resulting neighborhood graphs.

3.1 Neighborhoods for Move (D1), Rotate (D2), and Anisotropic Size-Neutral Deform (D4)

When moving one region over the other (i.e., changing its position while keeping the region's other properties the same), starting with *inside*, the relations traverse via *coveredBy*, *overlap*, and *meet* to *disjoint*. Likewise, starting with *contains*, the relations make a traversal via *covers*, *overlap*, and *entwined* to *embrace*. The complete *contains-embrace* traversal is, however, only possible if the containing region *A* is not only larger than the contained region *B*, but also larger than *B*'s complement. Otherwise, a movement from *overlap* to *entwined* is impossible. Likewise the *overlap* to *meet* transition as part of the *inside-disjoint* traversal is only possible if the contained region *A* is smaller than the containing region's complement. The reverse traversals—from *disjoint* to *inside* and from *embrace* to *contains*—only follow the same path if the two related regions *A* and *B* fulfill a size constraint: for the *disjoint-inside* traversal *A* must be smaller than *B*, while for the *embrace-contains* traversal *B* must be smaller than *A*. For both traversals, however, the first two movements are independent of the regions' sizes, so that *disjoint-meet-overlap* and *embrace-entwined-overlap* are unrestricted neighborhood sequences.

Starting with *equal*, any movement results in *overlap*, while moving any region from an *attach* relation will result in *overlap*. The reverse movements—starting at *overlap*—are again constrained, because the movement from *overlap* to *equal* can only occur if both regions have the same size, same orientation, *and* same shape. Complementarily, the movement from *overlap* to *attach* can only occur if both regions have complementary sizes, orientations, and shapes. None of the remaining 45 candidates are neighbors under movement. All six neighbors of *overlap* are constrained. Reversely, however, any transitions to *overlap* may occur independently of the relations' sizes. Therefore, this neighborhood graph needs to be considered a *directed* graph.

When rotating one of the regions (i.e., changing its orientation while retaining the region's other properties), exactly the same patterns of neighborhood transitions are obtained as for moving. This observation should not surprise since a rotation around a remotely located point has essentially the same observable effect as a movement.

The third type of deformation that exposes the very same neighborhoods is an anisotropic size-neutral deformation, which changes the region's shape, while preserving its size. An example of such a deformation is a simultaneous indenting and outdenting by the same amount. Starting at *equal*, an anisotropic size-neutral deformation results in *overlap*; likewise, starting at *attach* the same deformation yields *overlap*. The reverse deformations are only possible if both regions are of the same size (from *overlap* to *equal*) or if their sizes are equal to the other region's complement (from *overlap* to *attach*). Therefore, the neighborhoods from *overlap* to *equal* and to *attach* are constrained, while they are unconstrained in the reverse direction. Since this deformation has no impact on the region's size, it cannot yield any of {*coveredBy*, *inside*, *covers*, *contains*} from *equal*. Likewise, this deformation cannot change *attach* to any of {*entwined*, *embrace*}, because these two relations require *A*'s exterior to be a subset of *B*'s closure, while *attach* requires *A*'s exterior to be equal to *B*'s closure. For the converse reason (i.e., the closure of *A*'s exterior is a superset of *B*) none of {*disjoint*, *meet*} can result from a size-neutral deformation of *attach*. The relative sizes of the regions also imply that the four pairs of *disjoint-meet*, *inside-coveredBy*, *contains-covers*, and *entwined-embrace* are unrestricted neighbors in both directions. The two deformations from *overlap* to *entwined* and to *meet* need special attention. Since *entwined* requires *A*'s closure to be a superset of the closure of *B*'s exterior, a size-neutral deformation from *overlap* is only possible if *A* is larger than the closure of *B*'s exterior. Since the reverse deformation has no size constraint, the *overlap-entwined* neighborhood is as restricted as the *overlap-covers* and *overlap-coveredBy* neighborhoods. Finally, the transition from *overlap* to *meet* is only possible if *A* is smaller than *B*'s exterior, because otherwise *attach* (for same size) and *entwined* (for larger) would be obtained.

Since the three deformations share the same conceptual neighborhoods, we refer to this neighborhood as the MRAD (***m***ove, ***r***otate, and ***a***nisotropic size-neutral ***d***eform) neighborhood.

3.2 Neighborhoods for Isotropic Scale (D3)

An isotropic scaling either reduces or increases the size of an object by the same factor in all directions, leading for small scale changes to similar figures (i.e., objects with the same shapes, but of different sizes). The forming of a buffer zone around a region, either to its inside (reduce) or its outside, is essentially the result of an isotropic scaling. For convex regions, isotropic scaling changes the size of the object, but, if applied in isolation from other deformations, it has no effects on the object's shape, position, and orientation. Isotropic growing and isotropic shrinking are qualitative refinements of isotropic scaling, but since the coarsest categorization of deformations considers only *same* and *different*, these refinements are not considered here. Nevertheless, the scaling factor has an upper and a lower bound in order to guarantee that the region remains a region and neither subsumes the entire embedding space (i.e., S^2) nor collapses to a point, respectively.

Starting at *equal* isotropic scaling leads immediately, and only, to *inside* or *contains.* No other relations can be obtained from an isotropic scaling of *equal*. The reverse transitions are only possible if the two regions have the same shapes, positions, and

orientations. This same behavior applies to the isotropic scaling of *attach* with respect to *disjoint* and *embrace*, but the constraints (about same shape, position, and orientation) for isotropic scaling that starts at *embrace* apply now to a region's complement.

Starting at *contains*, the isotropic scaling (down) of the containing region leads from *contains* over *covers* to *overlap*. Depending on the position with respect to the contained region, further scaling (down) either leads via *coveredBy* to *inside* (Fig. 3a) or via *meet* to *disjoint* (Fig. 3b). The first transition (from *contains* to *covers*) is constrained as the two regions must have different positions, different orientations, *or* different shapes (otherwise the transition is made to *equal*). Similar transition paths are followed if the isotropic scaling (down) proceeds from *embrace* via *entwined* to *overlap*, where again a bifurcation occurs, either via *coveredBy* to *inside* or via *meet* to *disjoint*. The choice depends on whether the scaled region's center is located in the other regions' interior (path towards *inside*) or in that region's exterior (path towards *disjoint*). The constraint about the regions' different shapes, positions, or orientations (for the *contains*-to-*covers* scaling) repeats as constraints with the other region's complement.

(a) (b)

Fig. 3. Starting with *contains*, an isotropic scaling leads either to (a) *disjoint* or (b) *inside*, depending on the contained region's position in the containing region

If the isotropic scaling (up) starts at *inside* (*in lieu* of *contains*) and *disjoint* (*in lieu* of *embrace*) the transitions and patterns are mirrored along the eleven relations' vertical axis, replicating also the constraints.

3.3 Neighborhoods for Anisotropic Scale (D10)

Like isotropic scaling, anisotropic scaling reduces or increases the size of an object, however, the scaling factor is not the same in all directions. Two types of anisotropic scaling may occur, depending on whether the scaling is non-zero in all directions, or whether it is zero in some directions. If the scaling is non-zero in all directions *and* of the same factor, then this deformation is not anisotropic, but isotropic (Section 3.2). Also a zero-scaling in *all* directions would have no effect as does not change the size, yielding the neutral scaling. We refer to the ***a***nisotropic ***s***caling with a zero scale in some (but not all) direction(s) as AS-0, and to the ***a***nisotropic ***s***caling without zero scales as AS. Anisotropic scalings imply that the shape as well as the overall orientation may be affected, by the deformation, but neither change in orientation nor change in shape are intended with the this deformation.

3.3.1 Anisotropic Scale without Zero Scaling Factor

When compared with isotropic scaling IS, anisotropic scaling AS has two different impacts on conceptual neighbors. (1) While IS deforms neither *equal* nor *attach* to *overlap*, AS does so (Fig. 4c). (2) While the four IS transitions from *disjoint* (to *meet* and *attach*), from *inside* (to *equal* and *coveredBy*), from *contains* (to *equal* and *covers*), and from *embrace* (to *attach* and *entwined*) are constrained, the same AS transitions are unconstrained. Since the deformation from *equal* to *coveredBy* or *covers* would require some zero-scaling (where the boundaries coincide), AS cannot make that transition. The same restriction applies to the transitions from *attach* to *meet* and *entwined*.

3.3.2 Anisotropic Scale with Partial Zero Scaling Factor

The inclusion of at least some local zero-scaling in AS-0 implies that the boundary of the scaled region remains at its position in at least one place, but not everywhere. It is most noticeable for *equal* (and *attach*), where AS and AS-0 have no common neighbors: the two AS-neighbors of *equal*—*inside* and *contains* (Fig. 4a)—cannot be obtained with AS-0, because at least some part of the common boundary in *equal* needs to be shared with the relation of the deformed region, but *inside* and *contains* have empty boundary-boundary intersections [14]. On the other hand, the two AS-0 neighbors of *equal*—*coveredBy* and *covers* (Fig. 4b)—could not be realized for AS, because these transitions require a zero-scaling in some direction to maintain the boundary-boundary intersection. An AS-0 deformation from *overlap* to *equal* (and reverse) is possible if at least some boundary parts are identical between the two configurations, for instance, the nodes at which the boundaries cross (Fig. 4c). The corresponding deformations are possible for AS-0 when swapping *equal* with *attach*.

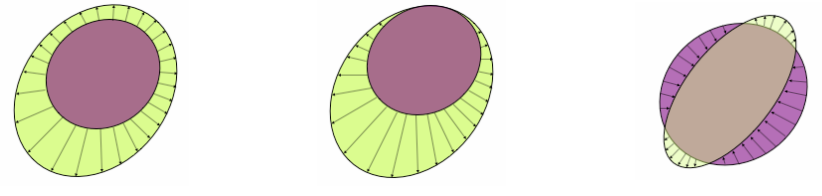


Fig. 4. Anisotropic scalings converting *equal* into (a) *inside* (for AS, but not AS-0), (b) *coveredBy* (for AS-0, but not AS), and (c) *overlap* (for AS and AS-0)

3.4 Conceptual Neighborhood Graphs for Simple Deformations

The transitions and their constraints discussed for the simple deformations yield four conceptual neighborhood graphs (Fig.5). Transitions that can only occur if the related regions fulfill a particular geometric constraint are unidirectional and, therefore, lead to directed edges. The five deformations analyzed give rise to four different neighborhood graphs. With ten out of 55 possible edges (i.e., 10%), the MRAD graph (Fig. 5a) is the least saturated neighborhood graph. The IS-neighborhood graph (Fig. 5b) has twelve *directed* edges (i.e., a 21% saturation), while the AS-neighborhood graph

(Fig. 5c) features 14 out of 55 possible edges (i.e., 25% saturation), with four of *overlap*'s neighbors being constrained. The AS-0 neighborhood graph (Fig. 5d) also has 14 edges, but all are undirected.

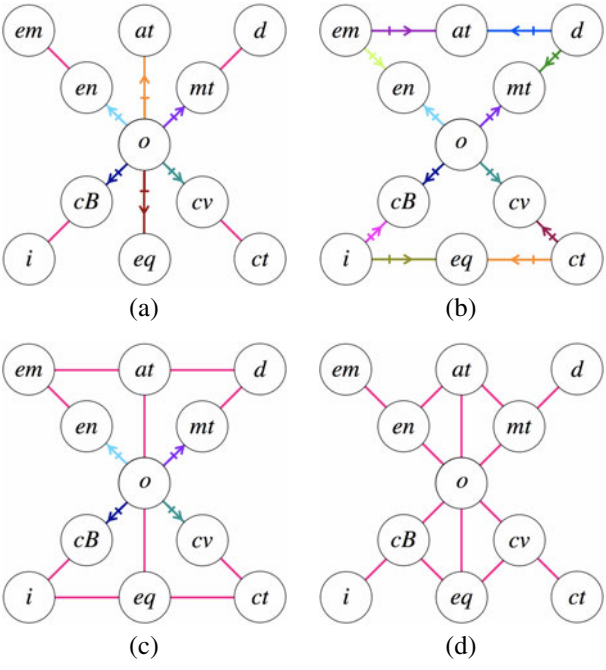


Fig. 5. The conceptual neighborhood graphs obtained from (a) moving, rotating, or anisotropic size-neutral deforming (MRAD); (b) isotropic scaling (IS); (c) anisotropic scaling (AS); and (d) anisotropic scaling with a zero-factor (AS-0)

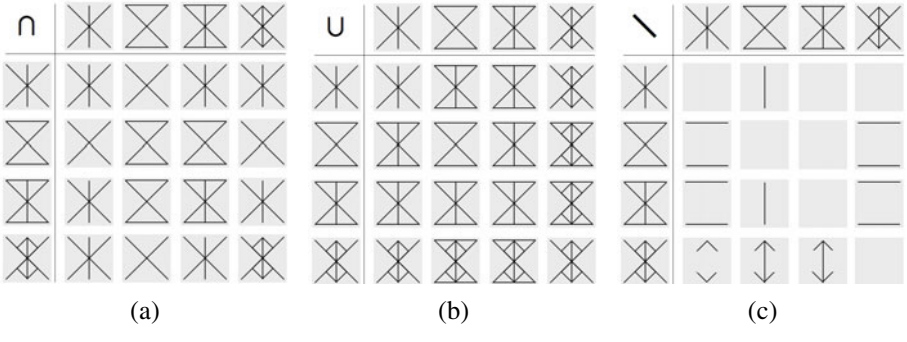


Fig. 6. Combinations of the MRAD, IS, AS, and AS-0 neighborhood graphs with (a) intersection, (b) union, and (c) difference

The combinations of the neighborhood graphs with intersection, union, and difference in an iconic form, disregarding any direction constraints (Fig. 6), shows that all intersections and unions feature one additional element each—⨯ and ⧖—while the difference has four additional elements. All other combinations result in one of the four basic neighborhood graphs. The additional intersection graph does not connect to *equal* and *attach*, so it only provides opportunities to relax nine of the eleven relations. The additional difference graphs are also specialized for subsets.

4 Comparisons with A-B-C Neighborhoods

The definition of the A-B-C neighborhoods [18] relied on lines embedded in R^1. A-neighbors result from moving one of the four endpoints of an interval, essentially simulating an anisotropic scaling (Fig. 7a); B-neighbors have a fixed duration, therefore, simulating an isolated movement (Fig. 7b); and C-neighbors result from expanding or shrinking an interval's duration intending to model an isotropic scaling (Fig. 7c).

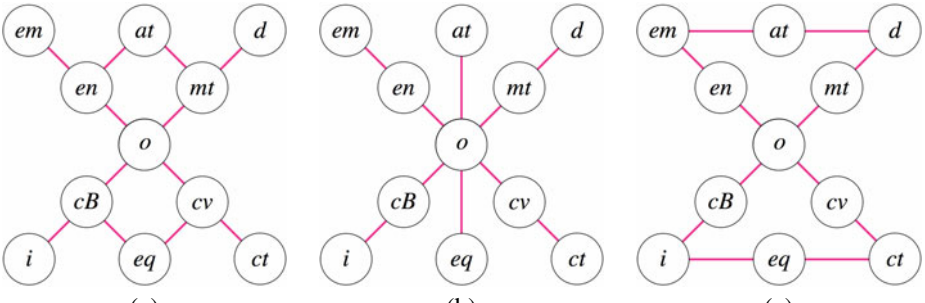


Fig. 7. The conceptual neighborhood graphs of the 11 region-region relations in S^2 for (a) A-neighborhood, (b) B-neighborhood, and (c) C-neighborhood

Compared with the graphs obtained for isolated deformations (Figure 5a-d), the MRAD-neighborhood is a refinement of the B-neighborhood as MRAD considers the constraints when deforming an *overlap* relation, yielding a directed graph, while the B-neighborhood is an undirected graph. Similarly, the IS-neighborhood is a refinement of the C-neighborhood as IS is a directed graph (without a single undirected edge), whereas the C-neighborhood graph is identical, but only with undirected edges.

The correspondences between the A-B-C neighborhoods and the two AS-neighborhoods are more complex, however. The intersections of the graphs show that there is no perfect match between the two AS-neighborhoods and any of the three A-B-C neighborhoods. However, the AS-0 neighborhood is a superset of the A-neighborhood and the B-neighborhood, and the AS-neighborhood is a superset of the B-neighborhood and the C-neighborhood. The strongest correlation appears between the AS-0 neighborhood and the A-neighborhood, because they only differ in the two edges from *overlap* to *equal* and to *attach*.

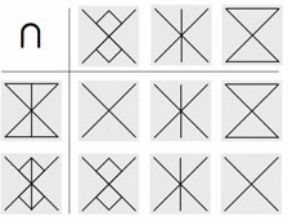


Fig. 8. Intersections of the AS and AS-0 graphs with the A-B-C neighborhood graphs

This similarity between AS-0 and the A-neighborhood was somehow expected, given that the A-neighborhood specification for intervals—fix one endpoint, move the other—is an anisotropic deformation with a zero-factor at one boundary point. So it is more the non-perfect match that surprises. The reason for the deviation is founded in the different natures of the boundaries for intervals and simple regions (i.e., a region that is homeomorphic to a 2-disk and, therefore, has no holes and no separations). A simple region's boundary is simply connected (forming a Jordan curve), while the interval's boundary is disconnected. For two *equal* intervals, an anisotropic scaling with a zero-factor can only keep one boundary point fix and can only move one (i.e., the other) boundary point. This implies that *equal* cannot be deformed into *overlap*, because this change would require that both boundary points move. On the other hand, the very same deformation applied to two *equal* regions can result in *overlap*, because the region's boundary is not a set of two disconnected points, but a connected line, part of which can remain put (i.e., keeping the intersection between the two boundaries), part of which can move into the other region's interior, and yet another part can move into the other region's exterior (Fig. 4c). Therefore, an anisotropic scaling with a zero-factor yields a different conceptual neighborhood for regions than for intervals. A similar difference between region and interval relations was already observed with the specification of the *overlap* relation in terms of the 4-intersection or 9-intersection [17] where two overlapping regions have a boundary-boundary intersection, while two overlapping intervals do not.

Since the A-neighborhood graph also coincides with the neighborhood derived from the least number of differences in the relations' 9-intersection matrices for region-region relations, both in R^2 [13] and in S^2, the redundancy would have been expected to be an indicator that the A-neighborhood is applicable across object types.

Finally the AS-neighborhood has no good match with any of the three A-B-C neighborhoods.

5 Relationships among Conceptual Neighborhoods

The extend collection of conceptual neighborhood graphs begs the question of how these graphs are interrelated, and whether there is a finite, complete set of neighborhood graphs. The ordering of the known graphs in the form of a lattice offers a structure with which these questions can be answered.

The set of neighborhood graphs to consider are (1) MRAD, (2) IS, (3) AS, (4) AS-0, and (5) A. We add to them the two additional graphs that were found among

the intersections and unions, yielding a set of seven graphs. The union of all graphs is the top element, while the intersection of all graphs—the two crossing diagonals—is the bottom element.

Above the bottom element are those three graphs that each add to the bottom element one set of traversals: MRAD adds the vertical line for the *equal* to *attach* traversal; IS adds the horizontal lines for the *inside-contains* and *disjoint-embrace* traversal; and the A-neighborhood adds the two chains *coveredBy-equal*-covers and *meet-attach-entwined*.

Each of these three graphs has two covering graphs, adding always one of the three traversals. Adding to MRAD the two chains *coveredBy-equal*-covers and *meet-attach-entwined* yields AS-0, while adding to it the two horizontal lines for the *inside-contains* and *disjoint-embrace* traversals gives rise to AS. Adding to the A-neighborhood the vertical line for the *equal* to *attach* traversal also generates AS-0, and AS is the result of adding to IS the vertical line for the *equal* to *attach* traversal. The final graph results both from IS (by adding the two chains *coveredBy-equal*-covers and *meet-attach-entwined*), and to the A-neighborhood the horizontal lines for the *inside-contains* and *disjoint-embrace* traversals. AS-0, AS, and the third graph are topped off with the union by adding to each one of the three incremental patterns.

Pairs of graphs consistently form unions upward, and intersections downwards (Fig. 9). No further graph can be generated with unions or intersections, providing the full family of conceptual neighborhood graphs. Four family members resulted from the deformation categorization; the three A-B-C neighborhoods are included as well; and apart from the top and bottom elements, one new graph—as the disjunction of A and C—shows up.

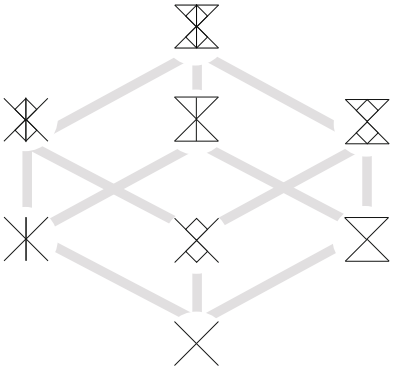


Fig. 9. The family of conceptual neighborhood graphs

6 Conclusions

Based on a categorization of deformation types—built from *same* and *different* positions, orientations, sizes, and shapes—we derived the conceptual neighbors from simple deformations. The targeted set of relations—the eleven binary topological relations between two regions on the sphere—highlighted repeatedly symmetric and converse

patterns, but also patterns due to symmetry with the complement of regions. The approach gave rise to four different neighborhood graphs, three of which are directed graphs due to geometric constraints on particular transitions between relations. The recognition of these constrained neighborhoods is a new insight. The undirected versions of two of the four graphs correspond to Freksa's B and C neighborhoods, but none of the other two graphs relate to the A-neighborhood. When organized in a lattice, with intersections downward and union upwards, a total of eight graphs are found that form the family of conceptual neighborhood graphs.

Acknowledgments

Discussions with Tony Cohn were enlightening as they helped clarify some of the aspects of conceptual neighborhoods. This work was performed during a sabbatical leave at the Instituto Nacional de Pesquisas Espaciais (INPE) in São José dos Campos, Brazil. Support by INPE is gratefully acknowledged.

References

1. Alexandroff, P.: Elementary Concepts of Topology. Dover, Mineola (1961)
2. Allen, J.: Maintaining Knowledge About Temporal Intervals. Communications of the ACM 26(11), 832–843 (1983)
3. Bruns, H., Egenhofer, M.: Similarity of Spatial Scenes. In: Kraak, M.J., Molenaar, M. (eds.) Seventh International Symposium on Spatial Data Handling (SDH 1996), vol. 4A, pp. 31–42 (1996)
4. Clementini, E., Di Felice, P.: An Algebraic Model for Spatial Objects with Indeterminate Boundaries. In: Burrough, P., Frank, A. (eds.) Geographic Objects with Indeterminate Boundaries, pp. 155–170. Taylor & Francis, Bristol (1996)
5. Cohn, A., Bennett, B., Gooday, J., Gotts, N.: Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. GeoInformatica 1(3), 1–44 (1997)
6. Cohn, A., Gotts, N.: The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. In: Burrough, P., Frank, A. (eds.) Geographic Objects with Indeterminate Boundaries, pp. 171–187. Taylor & Francis, Bristol (1996)
7. Cohn, A., Hazarika, S.: Qualitative Spatial Representation and Reasoning: An Overview. Fundamenta Informaticae 46(1-2), 2–32 (2001)
8. Cohn, A., Renz, J.: Qualitative Spatial Representation and Reasoning. In: van Hermelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, pp. 551–596 (2008)
9. Egenhofer, M.: Query Processing in Spatial-Query-by Sketch. Journal of Visual Languages and Computing 8(4), 403–424 (1997)
10. Egenhofer, M.: Deriving the Composition of Binary Topological Relations. Journal of Visual Languages and Computing 5(2), 133–149 (1994)
11. Egenhofer, M.: Spatial SQL: A Query and Presentation Language. IEEE Transactions on Knowledge and Data Engineering 6(1), 86–95 (1994)
12. Egenhofer, M.: Spherical Topological Relations. Journal on Data Semantics III, 25–49 (2005)

13. Egenhofer, M., Al-Taha, K.: Reasoning about Gradual Changes of Topological Relationships. In: Frank, A.U., Formentini, U., Campari, I. (eds.) GIS 1992. LNCS, vol. 639, pp. 196–219. Springer, Heidelberg (1992)
14. Egenhofer, M., Franzosa, R.: Point-Set Topological Relations. International Journal of Geographical Information Systems 5(2), 161–174 (1991)
15. Egenhofer, M., Herring, J.: A Mathematical Framework for the Definition of Topological Relationships. In: Brassel, K., Kishimoto, H. (eds.) Fourth International Symposium on Spatial Data Handling, pp. 803–813 (1990)
16. Egenhofer, M., Herring, J.: Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases, Department of Surveying Engineering, University of Maine, Orono, ME (1991)
17. Egenhofer, M., Sharma, J., Mark, D.: A Critical Comparison of the 4-Intersection and the 9-Intersection Models for Spatial Relations: Formal Analysis. In: McMaster, R., Armstrong, M. (eds.) Autocarto 11, Minneapolis, MD, pp. 63–71 (1993)
18. Freksa, C.: Temporal Reasoning based on Semi-Intervals. Artificial Intelligence 54(1), 199–227 (1992)
19. Freeman, J.: The Modeling of Spatial Relations. Computer Graphics and Image Processing 4(2), 156–171 (1975)
20. Galton, A.: Spatial and Temporal Knowledge Representation. Earth Science Informatics 2(3), 169–187 (2009)
21. Klippel, A., Li, R.: The Endpoint Hypothesis: A Topological-Cognitive Assessment of Geographic Scale Movement Patterns. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 177–194. Springer, Heidelberg (2009)
22. Kurata, Y., Egenhofer, M.: The Head-Body-Tail Intersection for Spatial Relations Between Directed Line Segments. In: Raubal, M., Miller, H.J., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2006. LNCS, vol. 4197, pp. 269–286. Springer, Heidelberg (2006)
23. Mark, D., Egenhofer, M.: Modeling Spatial Relations between Lines and Regions: Combining Formal Methods and Human Subjects Testing. Cartography and Geographic Information Systems 21(4), 195–212 (1994)
24. Randell, D., Cui, Z., Cohn, A.: A Spatial Logic based on Regions and Connection. In: Nebel, B., Rich, C., Swartout, W. (eds.) 3rd International Conference on Principles of Knowledge Representation and Reasoning KR 1992, pp. 165–176 (1992)
25. Randell, D., Cohn, A., Cui, Z.: Computing Transitivity Tables: A Challenge for Automated Theoremprovers. In: 11th International Conference on Automated Deduction (CADE 1992), pp. 786–790 (1992)
26. Retz-Schmidt, G.: Various Views on Spatial Prepositions. AI Magazine 9(2), 95–105 (1988)
27. Reis, R., Egenhofer, M., Matos, J.: Conceptual Neighborhoods of Topological Relations between Lines. In: Ruas, A., Gold, C. (eds.) The 13th International Symposium on Spatial Data Handling (SDH 2008), Montpellier, France, pp. 557–574. Springer, Heidelberg (2008)
28. Schlieder, C.: Reasoning About Ordering. In: Kuhn, W., Frank, A.U. (eds.) COSIT 1995. LNCS, vol. 988, pp. 341–349. Springer, Heidelberg (1995)
29. Schneider, M., Behr, T.: Topological Relationships between Complex Spatial Objects. ACM Transactions on Database Systems 31(1), 39–81 (2006)

Detecting Road Intersections from GPS Traces

Alireza Fathi[1] and John Krumm[2]

[1] College of Computing
Georgia Institute of Technology
Atlanta, Georgia USA
alireza.fathi@gmail.com
[2] Microsoft Research
Microsoft Corporation
Redmond, Washington USA
jckrumm@microsoft.com

Abstract. As an alternative to expensive road surveys, we are working toward a method to infer the road network from GPS data logged from regular vehicles. One of the most important components of this problem is to find road intersections. We introduce an intersection detector that uses a localized shape descriptor to represent the distribution of GPS traces around a point. A classifier is trained on the shape descriptor to discriminate intersections from non-intersections, and we demonstrate its effectiveness with an ROC curve. In a second step, we use the GPS data to prune the detected intersections and connect them with geometrically accurate road segments. In the final step, we use the iterative closest point algorithm to more accurately localize the position of each intersection. We train and test our method on GPS data gathered from regular vehicles in the Seattle, WA, USA area. The tests show we can correctly find road intersections.

Keywords: GPS, road map, road network, intersection detection.

1 Introduction

Digital road maps are clearly important for both consumers and businesses. At present, these maps are created by companies fielding fleets of specialized vehicles equipped with GPS to drive the roads and record data. This is an expensive process, and it is difficult to keep up with changes in the road network. An emerging alternative is to use GPS data from regular vehicles driving their regular routes. This has the advantage of easily scaling to the entire road network and providing much more up-to-date data whenever roads change.

The challenge of this technique is how to process all the data into a road map. The OpenStreetMap [1] project provides one model where volunteers manually edit GPS traces and aerial images into digital maps. While OpenStreetMap moves away from the use of specialized vehicles, we would like to eliminate the manual step. In this paper, we show how to automate one important aspect of this processing: finding road intersections. We test our process on a large amount of GPS data we gathered from

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 56–69, 2010.

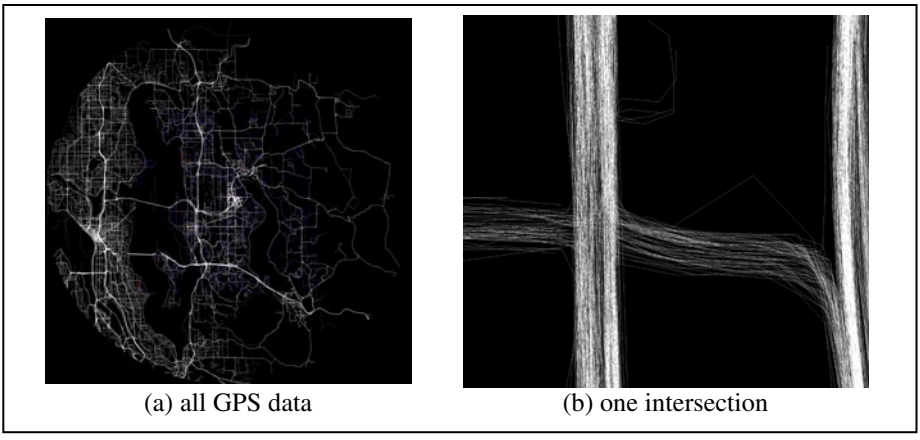


Fig. 1. GPS traces. (a) shows an overview of our GPS traces from the greater Seattle, WA USA area. In (b) is a close-up of traces around a road intersection.

vehicles that were already driving in our metropolitan area. This data is shown in Fig. 1(a).

Our algorithm detects intersections in the GPS data, an example of which is shown in Fig. 1(b). It begins by using a shape descriptor trained on positive and negative examples of intersections. Next it connects the intersections by finding vehicle traces that move between them. Finally, the algorithm refines the locations of the intersections based on the GPS data associated with the nearby roads. We evaluate our algorithm by comparing it to a known road network. Specifically, we evaluate it in terms of its ability to find intersections, the accuracy of the intersections' computed locations, and the accuracy of the lengths of the roads between the intersections.

2 Previous Work

Some of the earliest research into building road maps was based on aerial images, such as that by Tavakoli & Rosenfeld [2]. They group together edges found by an edge detector into shapes representing buildings and roads. Hu *et al.* [3] find seed pixels that were likely roads and then grew from the seeds by tracking along road segments. Barsi & Heipke [4] trained a neural network to find intersections in images. This is related to our work in that we also train an intersection detector, although ours is formulated differently and works on GPS traces rather than image pixels.

Another line of research addresses the problem of refining an existing map using GPS traces. Rogers *et al.* [5] use an initial map to refine the centerline of the road. In this work, they look at perpendiculars to the refined centerline and cluster traces into lanes. Guo *et al.* [6] present initial simulation work with a similar goal of finding the centerline of the road. Our goal is to build a road map without any prior road map.

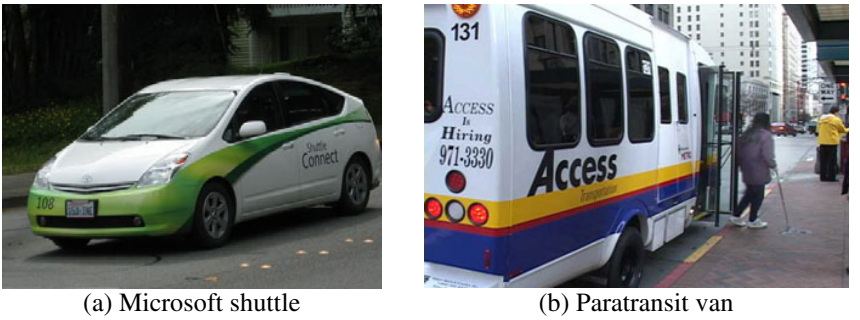
There are other efforts with the same goal. Two of these, Brüntrup *et al.* [7] and Worral & Nebot [8] present simple clustering techniques for determining the location of the road. Some of the deepest work on this problem comes from Schroedl *et al.* [9]

and Edelkamp & Schrodel [10]. Starting with a blank map, they first find centerlines by clustering. Then they determine the structure of the lanes and intersections. Recent work by Cao and Krumm [11] builds a routable road network by first clarifying the GPS traces and then clustering them into a connected graph representing the roads.

Our approach is different than the above in that we begin by finding intersections using a detector trained on ground truth data. As far as we are aware, this represents the first use of a trained detector to find road features from GPS data. After finding intersections, we discover connecting edges by looking for contiguous traces going between them. We use these connecting edges to refine the location of the detected intersections. Before we discuss our technical approach, however, we present in the following section the GPS data we used for our training and testing.

3 GPS Data

We have collected GPS data for testing by deploying GPS loggers on 55 Microsoft Shuttles and 252 King County Paratransit vehicles, as depicted in Fig. 2. The GPS loggers we used for our experiments were RoyalTek RBT-2300 models using the SiRF Star III chipset with WAAS enabled. These are relatively accurate and provide data with a standard deviation of about 4.1 meters according to our experiments.



(a) Microsoft shuttle (b) Paratransit van

Fig. 2. A Microsoft shuttle is shown in (a), and a King County paratransit van is shown in (b)

The Microsoft shuttles roam around Redmond, Washington, USA, continuously during the day, and the paratransit vehicles move around Seattle, Washington, USA when they are called for service pickups.

Microsoft Shuttles: Shuttles provide both fixed and on-demand service between Microsoft campus buildings during the day. The GPS loggers mounted on these shuttles record time-stamped latitude/longitude coordinates with an interval of one second. We collected the recorded data over two weeks. From each shuttle we retrieved an average of about 358,300 time stamped latitude/longitude pairs, which corresponds to about 99.5 hours of data from each of the 55 shuttles. Most of the shuttles automatically turned off logging when they were parked for the night.

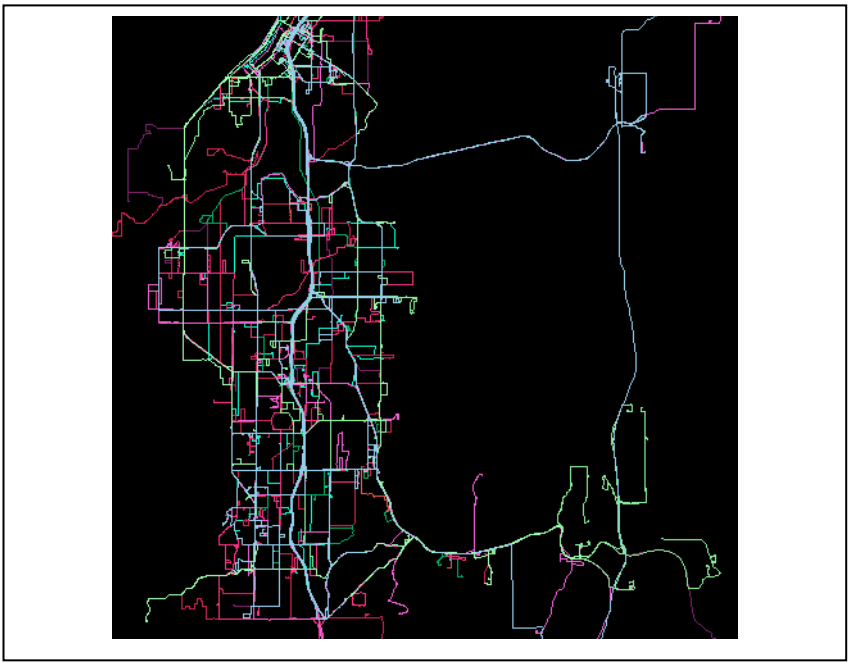


Fig. 3. This image shows trips segmented from our GPS data in different colors

King County Paratransit Vehicles: Paratransit vehicles provide on-demand services in Seattle and surrounding areas during the day and night. The GPS loggers mounted on these vehicles record time-stamped latitude/longitude coordinates with an interval of 5 seconds. We collected the recorded data over four weeks. From each vehicle we got an average of about 319,596 time stamped latitude/longitude pairs, which corresponds to about 444 hours of data from each of the 252 paratransit vehicles. The paratransit vehicles recorded continuously, which caused their logger memories to fill even when parked for the night. This is why we increased the sampling period to five seconds for these vehicles, as opposed to the shuttles' one second sampling period. Other than the sampling interval, the GPS loggers were identical for all the vehicles in our study. The two sets of vehicles have overlapping service areas, so we combined their data into one large data set.

As our first processing on this data, we split the traces into individual trips. Each trip consists of a sequence of GPS points which start when the vehicle starts moving and stops when the vehicle stops. We split the data into trips by looking at the gaps between timestamps to find when the vehicle was turned off. However, as mentioned previously, some loggers did not shut off automatically while the vehicles were parked, which provides us data while they were idle. To remove this useless data, we put a constraint on the minimum speed of the vehicle. Also sometimes we get very noisy latitude/longitude points from the loggers. To get rid of the noisy data, we also put a constraint on the maximum speed of the vehicles. Specifically, we split the data into trips whenever we find a gap of at least 10 seconds or the speed is slower than 5

miles per hour (mph) or faster than 90 mph. Retaining only the trip data means we can ignore data for parked vehicles and outliers, leading to faster downstream processing. Fig. 1(a) shows all the GPS trips for the Redmond and Seattle area in our database. We have depicted trips with different colors on a small area on the map in Fig. 3.

We use this data for detecting the locations of intersections. Having detected the intersections, we find the roads that connect them, and use these to refine the locations of the intersections. The next section describes how we detect intersections, followed by sections on refining their locations.

4 Intersection Detection

Our goal is to detect road intersections in GPS data. In a graph representation of the road network, intersections are nodes which are connected by road segments that serve as edges. An intersection is a location where more than two edges connect to each other. We describe in this section how it is possible to determine if an arbitrary location on the map is an intersection or non-intersection by looking at the local GPS traces around that point. After finding the intersections, we connect them based on information provided by the GPS traces. Then we refine the locations of the intersections. This section describes the first step of this process, which is detecting the intersections. Section 4.1 describes our detection process generally, and Section 4.2 gives details and results.

4.1 Intersection Detection Algorithm

Our intersection detector works by sliding a specialized, 2D circular window, called a local shape descriptor, over the GPS data to find places whose GPS data indicates an intersection. The shape descriptor describes the distribution of GPS points at each location on the map. The goal is to find a shape descriptor that (1) can perfectly discriminate between intersections and non-intersections and (2) can be represented as a feature vector which lets us apply machine learning to develop a classifier out of the positive and negative examples in our training data.

The basic idea behind the shape descriptor is illustrated in Fig. 4. Given a set of GPS edges (two temporally adjacent GPS points from the same vehicle), the shape descriptor captures the local distribution and direction of GPS edges in the circle around that location. Our shape description is a set of annular sections, each of which can be thought of as a histogram bin. For every given edge, we add a point to each bin that the edge passes through.

We map the bins of each shape descriptor to a vector and then learn a classifier out of all the feature vectors provided by the training examples. To map the shape descriptor to the feature vector, there are a few issues that we must consider. First, we normalize the raw counts in the bins so they sum to one. This helps neutralize the differences between heavily traveled and lightly traveled intersections.

The other issue is that the shape descriptors are sensitive to the headings of the roads converging at the intersection. In our training, we want to limit the apparent diversity of intersections, so we attempt to compute a canonical orientation for each intersection and then rotate the corresponding shape description so its canonical

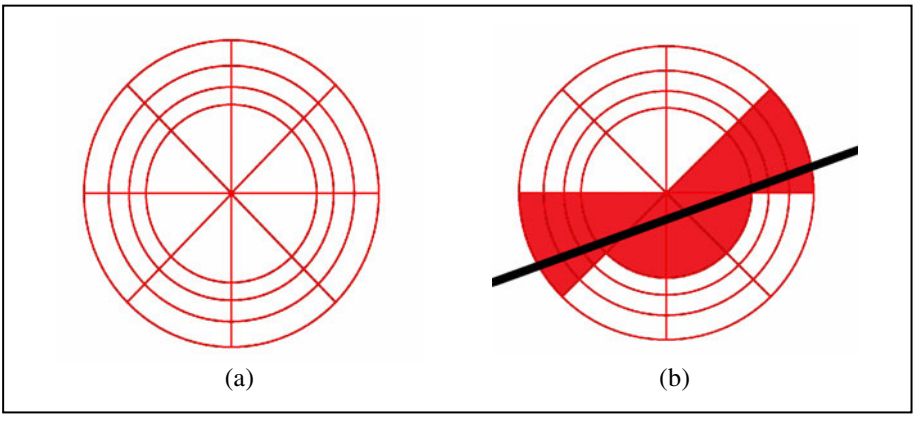


Fig. 4. The shape descriptor used for discriminating between intersections and non-intersections. (a) An example of shape descriptor with 32 bins. (b) For each edge passing from shape descriptor, a point is added to each of the bins the edge is passing through. The actual shape detector we used had each annulus split into 16 annular sections.

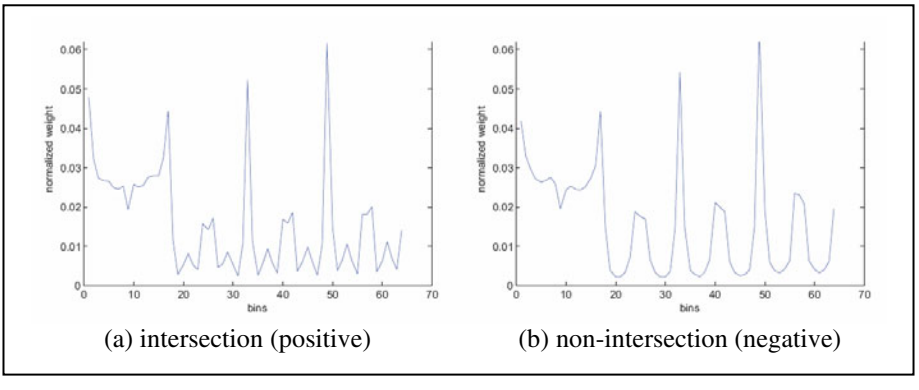


Fig. 5. Average of the 64 positive feature vectors is shown in (a) and average of 64 negative feature vectors is shown in (b). These are taken in annulus-major order, so bins 1-8 correspond to the inner annulus. As can be seen by comparing the plots, the positive features are more jagged. Because the negative samples are usually on straight roads, they have a peak every 180 degrees (every 8 bins), while the positive samples, which are 3- or 4-way intersections, have a peak usually every 90 degrees (every 4 bins).

orientation is zero. This helps make the intersection detector rotation invariant. We take the canonical orientation for each shape descriptor as the direction of the bin that has the maximum weight, and then we rotate the shape descriptor so this angle is zero. Then we insert the weights into the feature vector from those bins. In Fig. 5 we have shown average positive and negative examples of feature vectors.

We use the information provided by an available map to extract the ground truth location of intersections and the roads connecting the intersections to each other. Our

training data consists of positive examples taken from the locations of known intersections and negative examples taken along known roads that are at least 20 meters from any intersection. We look at all intersection types from our ground truth map, so our detector is not limited to any certain type of intersection.

To learn a classifier from the set of positive and negative feature vectors, we use the Adaboost algorithm, similar to that of Viola and Jones [12]. The Adaboost training algorithm learns a classifier as a subset of a group of weak classifiers that each can barely classify the data better than random. Adaboost adds the weak classifiers incrementally one at a time to the final classifier. At each iteration, the weight of each incorrectly classified example is increased, and the weak classifier that performs best on those is chosen to be added to the final classifier.

We consider the elements of the feature vector (*i.e.* normalized bins of the histogram) as potential weak classifiers $\{f_p\}$ for the Adaboost run. In each iteration t of the Adaboost training algorithm, one of the features $f_t \in \{f_p\}$ is chosen as the feature of the weak classifier $h_t(x)$ to be added to the final classifier. This weak classifier is of the form:

$$h_t(x) = \begin{cases} 1 & \text{if} \quad p_t f_t < p_t \theta_t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\theta_t \in (-\infty, \infty)$ is the classification threshold of the classifier and $p_t = \pm 1$ is a parity for the inequality sign. The parity allows either $f_t(x) < \theta_t$ or $f_t(x) > \theta_t$ as a potential feature.

After all T iterations of the algorithm, we get the final classifier $H(x)$, which is of the form

$$H(x) = \begin{cases} 1 & \text{if} \quad \sum_{t=1}^{T} \alpha_t h_t(x) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where α_t is the selected weight for weak classifier $h_t(x)$, as chosen by the Adaboost algorithm. The final classifier consists of a sum of the chosen weak classifiers. We can replace ≥ 0 in Equation (2) with $\geq \lambda$ to adjust the bias of the classifier to make it more or less sensitive at the cost of more false positives or more false negatives.

After learning the classifier that can discriminate between the intersections and non-intersections, we test it by moving it over the GPS traces. (Note that the training locations are at ground truth locations of intersections and roads, while the test locations are taken at the actual GPS measurements.) For every location we check, we fill the shape descriptors bins, normalize, rotate to a canonical orientation, and pass the feature vector through our classifier. The classifier $H(x)$ returns a number in the range of (−1, 1) for each test location. The locations with a greater value will have a higher probability of being an intersection. For each of the tested locations, we look at the value of the other locations around it. If there was a location nearby with a higher value, we discard the current location. We do the same for all the locations and prune the ones which are not a local maximum. This helps eliminate tight clusters of intersections. We accept the remaining locations as intersections if their classification value $H(x)$ is higher than a threshold λ.

We believe this is the first application of machine learning and a shape descriptor to the problem of finding intersections from GPS data.

The following section describes the specifics of our implementation of this approach to detecting intersections along with performance results.

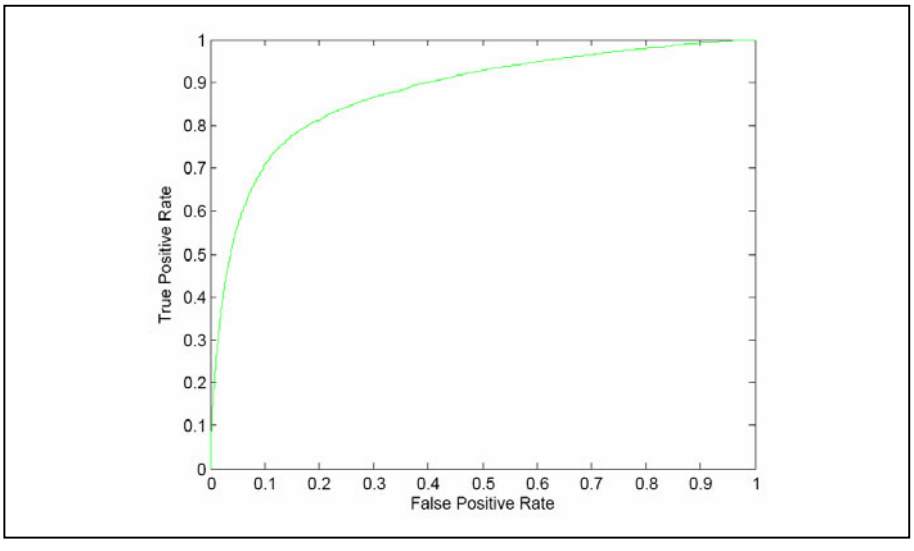


Fig. 6. This ROC curve shows the tradeoff between true positives and false positives for intersection detection with different values of the classifier threshold λ. The ideal operating point would be in the upper left corner with all true positives and no false negatives.

4.2 Intersection Detection Specifics and Performance

The shape descriptor has a few parameters that we need to tune, which are the number of circles, the radius of the smallest circle, the ratio between the radii of circles, and the number of angular slices. To discover the best set of parameters, we sweep through these parameters, training a classifier on half of the training data and measuring performance on the other half. The following parameters for the shape descriptor yielded the best performance: 4 circles, smallest radius of 20 meters, ratio between radii of circles of 1.3, and 16 angular slices.

Using the shape descriptor parameters just discovered, our next step is to learn a classifier using all the training data. The training data consists of positive examples which are shape descriptors centered at ground truth locations of known intersections. However, we do not have GPS data for all the area, because some intersections were never visited by our GPS loggers. Also, for some of the visited intersections, there is not data available for all roads connected to that intersection. We prune these before learning the classifier. The negative examples are taken from shape descriptors put on the ground along known roads at least 20 meters from any intersection.

We learn a classifier using 2000 iterations of the Adaboost algorithm. During the test, we look at all the GPS data. We sample points on the GPS traces, taking a point

as a sample if there is not a point sampled already within 5 meters. After filling the shape descriptor for a sampled point, we apply our classifier to measure the likelihood of the point being an intersection. The classifier assigns a value between −1 and 1, which we threshold with the value λ to determine whether or not the point is an intersection. The performance of the classifier is shown in Fig. 6 using an ROC curve that demonstrates performance for different values of the threshold λ. If a point is within 10 meters of a ground truth node, we declare that it should be labeled positive, otherwise it should be labeled negative. Section 6 describes how we refine the locations of these intersections.

We remove the points that have another point with a higher classifier output within 20 meters or whose classifier output value is less than $\lambda = 0.03$. This replaces multiple positive responses close to an intersection with a single detected intersection.

The next step in our overall process is to find roads that run between the intersections we found, which we describe in the next section.

5 Finding Roads

With intersections found, we next find roads that connect the intersections. While our main goal in this paper is to detect intersections, discovering the roads between intersections is a necessary prerequisite for our final step of refining the intersections' locations. This section describes how we filled in the roads between intersections and presents quantitative results comparing the lengths of our discovered roads to their true lengths.

5.1 Road Finding Algorithm

Each trip in our database represents a vehicle moving from one location to another. We use the trips to connect the intersections to each other. Each trip may pass through a number of intersections.

For each trip, we find the intersections that the trip passes through. These intersections are the ones that are near the path taken by the vehicle on that trip. We sort these intersections based on their path distance from the beginning of the trip, and we connect each intersection to its adjacent ones in the trip. While many trips from various vehicles may connect two adjacent intersections, we choose the one trip with the minimum distance between the two intersections as the best one.

This algorithm may cause some mistaken connections. Assume intersection A is connected to intersection C through intersection B. However, a trip might be noisy and may not pass close enough to B. This causes our algorithm to connect A to C, which is not the case in ground truth. To prune such mistakes, for every two connected nodes M and N, we find their shortest indirect path, *i.e.* the path that connects them through another intersection. If the shortest indirect path is almost the same length as the direct path, we remove the direct path. This is because, in this case, the direct path likely missed an intervening intersection slightly.

Another mistake we correct is some of the false positive intersections we found based on the shape descriptor method in Section 4.1. These false intersections will have only two roads connected to them, which violates our definition of an intersection. Thus, we prune all intersections that have two or fewer connected roads.

Fig. 7. After connecting the intersections using the road filling algorithm, there are some mistaken connections. In (a) we have shown the road network before pruning incorrect connections. In (b) the road network is depicted after removing incorrect connections.

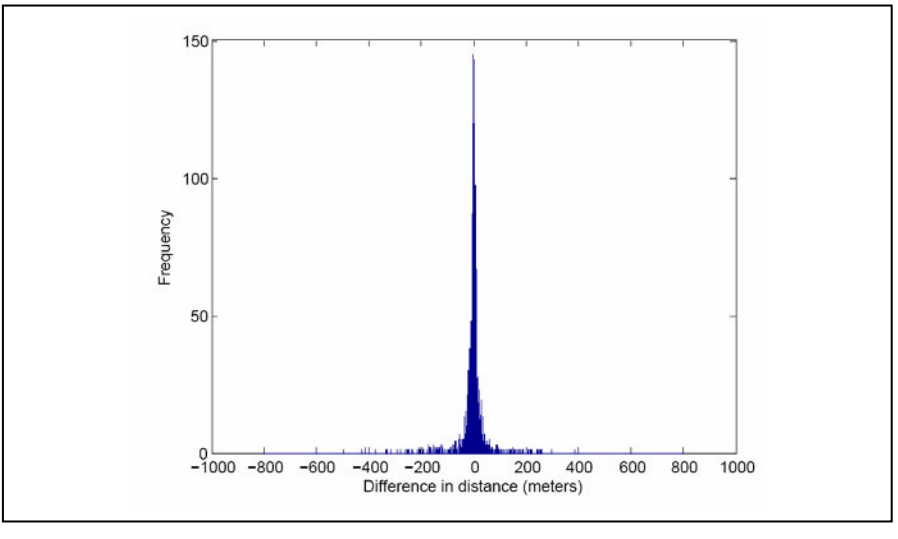


Fig. 8. This is the histogram of differences between path distance in the computed road network and ground truth. The errors are tightly clustered around zero.

5.2 Road Finding Specifics and Results

The result of road filling algorithm is presented in Fig. 7. If a trip passes closer than 30 meters to an intersection, it is assumed to be passing through that intersection. We remove a road between intersections A and node C with length l if there is another path between A and C with a length less than $\sqrt{2}l$. We remove the direct connection in this case because it means there are other intersections between A and C. We also remove an intersection if there are less than three roads connected to it. We see from Fig. 7 that the pruning technique effectively removes many mistaken roads. To measure the performance of road filling step, we compare the path distance between corresponding intersections in the computed road network to ground truth. We find the path distance between two intersections using the classic Dijkstra algorithm. In Fig. 8 we have shown the histogram of differences between path distances of

corresponding intersections in the computed road network and ground truth. The errors are tightly centered around zero.

These roads are valuable for refining the locations of the detected intersections. We discuss this process in the next section.

6 Refining Intersection Locations

In Section 4 we found the intersections and in Section 5 we connected the intersections with roads. The location of the intersections may be mistaken by a few meters, since we apply our intersection detector on a grid with 5 meter spacing. In this section we present an algorithm based on Iterative Closest Point (ICP) [13] to optimize the locations of the intersections.

6.1 Intersection Refinement Algorithm

The ICP algorithm is used to match two clouds of points. It iteratively matches the points in the model and the ones in the data, and it estimates the rigid transformation (translation and rotation) that moves the model to the data. In our specific case, the data is the raw GPS traces. The model is the detected intersection and the set of roads we found in Section 5 that connect to the detected intersection. Around each intersection we sample points on the GPS traces and project them on the roads connected to that intersection. Each sampled GPS point corresponds to its projection on the local road network around the intersection. Then we find a transformation matrix that maps the points on the model to the data points. We transform the intersection and the connected roads with this matrix. We keep iterating this procedure until convergence. The algorithm is as below:

1. Associate each point in the GPS traces (D) to its nearest neighbor in the road network (X) discovered in Section 5. D and X are $3 \times N$ matrices. Column i of D is a homogenous coordinate point in the data corresponding to column i of X, which is a homogenous point in the model. (A homogeneous coordinate has a 1 appended at the end, *e.g.* $(x, y, 1)^T$.)
2. Estimate the transformation matrix using the standard minimum squared error criterion:

$$T^* = \underset{T}{\operatorname{argmin}} |D - TX|_2 = DX^T(XX^T)^{-1}$$

3. Transform the model (*i.e.* intersection and connected roads) using the estimated transformation matrix: $X_{new} = T^* X_{previous}$.
4. If X is moved significantly, go back to step 1, otherwise terminate.

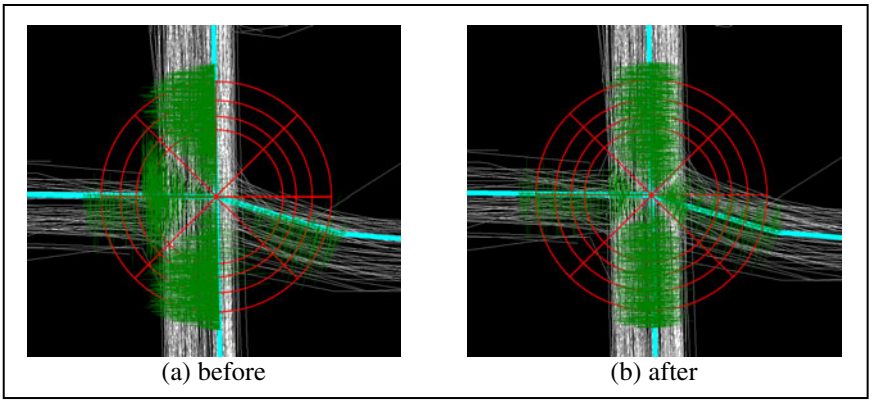


Fig. 9. The result of applying the ICP algorithm to optimize the node locations. In (a) the estimated node is a few metersoff from the ground truth. In (b) the location of the node is fixed after the ICP algorithm is applied.

In Fig. 9 we have shown the result of applying the ICP step to an intersection which is a few meters off from where it is supposed to be based on the GPS traces. The white lines represent the GPS traces, the red circle is the shape descriptor, the cyan edges show the road network around the intersection, and the green lines connect the corresponding points on the data and model.

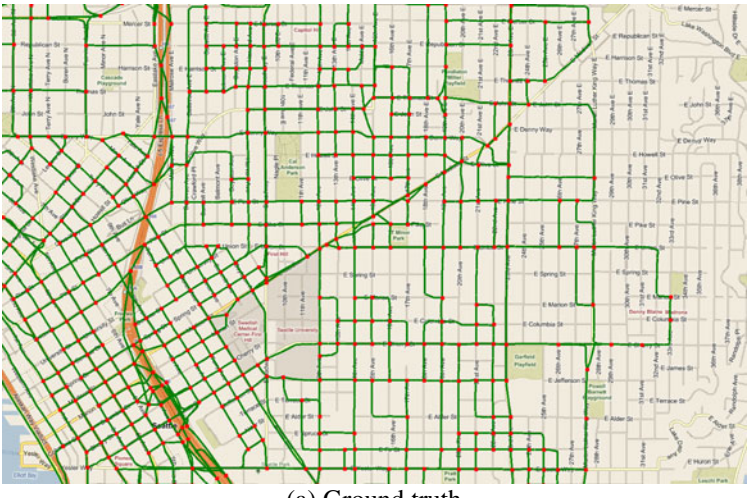
After we optimize the location of the intersections using the ICP algorithm, we next recompute the edges between the intersections. We perform the road filling algorithm described in Section 5 to recalculate the roads. We iteratively optimize the intersection locations using the ICP and recalculate the roads until convergence.

6.2 Intersection Refinement Results

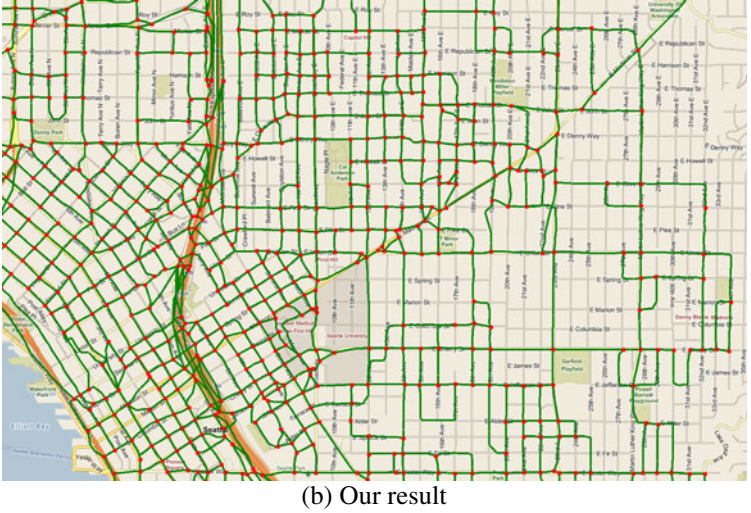
For each intersection, we find the correspondence between the GPS data around it and its connecting roads. We consider the local GPS points in an annulus with inner radius 25 meters and outer radius 40 meters, centered on the intersection. We do not consider the GPS points closer than 25 meters since they fall inside the intersection and make it difficult to find the correct corresponding connected road.

We applied the ICP algorithm to all our detected intersections that were within 20 meters of a ground truth intersection, which amounted to 76% of the ground truth intersections. After applying ICP, the average distance between the detected intersections and their ground truth locations improved from 7.2 meters to 4.6 meters, indicating that our refinement approach had a significant positive impact on the accuracy of the intersections' locations. Part of the remaining error could be due to inaccuracies in the underlying map.

We have compared the final results of our approach with the ground truth in Fig. 10.



(a) Ground truth



(b) Our result

Fig. 10. (a) Ground Truth image of the intersections and the roads connecting them. (b) The results computed by our algorithm. We remove from the ground truth the intersections and roads for which we do not have enough GPS data.

7 Conclusions

In this paper we have demonstrated a new pattern recognition method for finding road intersections based on GPS traces of everyday drivers. Our approach is an alternative to the costly and slow method of using dedicated drivers and vehicles for the purpose of map generation.

We find intersections from GPS data in three steps. First we find the intersections using a classifier learned over the shape descriptors. In the next step we connect the intersections with roads. In the last step we optimize the estimated road network to best fit the connected roads.

Future work should include an assessment of what types of intersections are most often found or missed with our approach, in particular highway merges where the exact intersection point is not as clear as a regular surface road intersection.

References

1. Haklay, M., Weber, P.: OpenStreetMap: User-Generated Street Maps. In: IEEE Pervasive Computing, pp. 12–18 (2008)
2. Tavakoli, M., Rosenfeld, A.: Building and Road Extraction from Aerial Photographs. IEEE Transactions on Systems, Man, and Cybernetics SMC-12(1), 84–91 (1982)
3. Hu, J., et al.: Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints. IEEE Transactions on Geoscience and Remote Sensing 45(12), 4144–4157 (2007)
4. Barsia, A., Heipkeb, C.: Artificial Neural Networks for the Detection of Road Junctions in Aerial Images. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Munich, Germany, pp. 113–118 (2003)
5. Rogers, S., Langley, P., Wilson, C.: Mining GPS Data to Augment Road Models. In: International Conference on Knowledge Discovery and Data Mining, pp. 104–113 (1999)
6. Guo, T., Iwamura, K., Koga, M.: Towards High Accuracy Road Maps Generation from Massive GPS Traces Data. In: IEEE International Geoscience and Remote Sensing Symposium, pp. 667–670 (2007)
7. Bruntrup, R., et al.: Incremental Map Generation with GPS Traces. IEEE Intelligent Transportation Systems, 574–579 (2005)
8. Worrall, S., Nebot, E.: Automated Process for Generating Digitised Maps through GPS Data Compression. In: Australasian Conference on Robotics and Automation, Brisbane, Australia (2007)
9. Schroedl, S., et al.: Mining GPS Traces for Map Refinement. Data Mining and Knowledge Discovery 9(1), 59–87 (2004)
10. Edelkamp, S., Schrödl, S.: Route Planning and Map Inference with Global Positioning Traces. In: Klein, R., Six, H.-W., Wegner, L. (eds.) Computer Science in Perspective. LNCS, vol. 2598, pp. 128–151. Springer, Heidelberg (2003)
11. Cao, L., Krumm, J.: From GPS Traces to a Routable Road Map. In: 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPAITLA GIS 2009), pp. 3–12. ACM, Seattle (2009)
12. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Computer Vision and Pattern Recognition (CVPR 2001), p I-511– I-518 (2001)
13. Besl, P.J., McCay, N.D.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)

Semantic Referencing – Determining Context Weights for Similarity Measurement

Krzysztof Janowicz[1], Benjamin Adams[2], and Martin Raubal[3]

[1] Department of Geography, The Pennsylvania State University, USA
jano@psu.edu
[2] Department of Computer Science, University of California, Santa Barbara, USA
badams@cs.ucsb.edu
[3] Department of Geography, University of California, Santa Barbara, USA
raubal@geog.ucsb.edu

Abstract. Semantic similarity measurement is a key methodology in various domains ranging from cognitive science to geographic information retrieval on the Web. Meaningful notions of similarity, however, cannot be determined without taking additional contextual information into account. One way to make similarity measures context-aware is by introducing weights for specific characteristics. Existing approaches to automatically determine such weights are rather limited or require application specific adjustments. In the past, the possibility to tweak similarity theories until they fit a specific use case has been one of the major criticisms for their evaluation. In this work, we propose a novel approach to semi-automatically adapt similarity theories to the user's needs and hence make them context-aware. Our methodology is inspired by the process of georeferencing images in which known control points between the image and geographic space are used to compute a suitable transformation. We propose to semi-automatically calibrate weights to compute inter-instance and inter-concept similarities by allowing the user to adjust pre-computed similarity rankings. These known control similarities are then used to reference other similarity values.

Keywords: Semantic Similarity, Geo-Semantics, Information Retrieval.

1 Introduction and Motivation

Similarity and analogy based reasoning are major approaches for the understanding of human cognition [1], work on artificial intelligence [2], as well as information retrieval and knowledge organization. In his classic book *Gödel, Escher, Bach - An Eternal Golden Braid*, Hofstadter, for instance, lists among the fundamental building blocks of human intelligence the ability *to find similarities between situations despite differences which may separate them [and] to draw distinctions between situations despite similarities which may link them* [3, p. 26]. The power of similarity lies in providing a graded structure instead of a rigid Boolean matching. In contrast to many purely syntactical or statistical

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 70–84, 2010.

measures, semantic similarity computes proximity based on the meaning of compared terms. Semantic similarity measures have a long tradition in GIScience – partially due to the analogy between measuring distances in geographic space and computing semantic similarity as inverse distance within a semantic or conceptual space [4]. Over the years, these measures have been applied to compute the similarity between spatial scenes [5,6], to improve landmark-based navigation [7], browsing through digital gazetteers [8], to support the classification of remote sensing data [9], or as additional reasoning service to compare or align classes, instances, and terms on the (geospatial) semantic Web [10,11,12,13,14]. The meaning of terms, however, is influenced or even determined by the context in which they are uttered. Therefore, meaningful similarities cannot be determined without taking additional contextual information into account [15,16,17]. A classical approach to make similarity theories context-aware is by introducing flexible weights. Most existing approaches to determine these weights are either too broad, application specific, or do not take users and their requirements into account [16]. In the past, the ability to adjust similarity theories until they fit a specific purpose has been one of the major criticisms for their evaluation.

In this work, we introduce a novel, context-aware, and semi-automatic weighting approach to better approximate the user's needs. The proposed methodology is inspired by the process of georeferencing images in which known control points between an image and the geographic space determine the appropriate transformation. In analogy, we propose to calibrate weights by allowing users to adjust the similarity of prominent pairs in a ranking. Some of these adjustments can be done automatically, for instance, by taking the user's location into account. Based on the user's *control similarities*, we can adjust the weights and hence reference other similarity values within an ontology. In analogy to geo-referencing, we call this process *semantic referencing*. Note, however, that in fact our approach is an optimization task.

The remainder of this paper is structured as follows. First we introduce related work on semantic similarity and information retrieval. Then, we discuss the relation between feature-based and geometric theories from on ontological perspective. Next, we introduce the theory of semantic referencing and diagnosticity measures. We then demonstrate our approach and its limits using an example from forestry and conclude the paper by pointing to open research questions.

2 Related Work

This section introduces related work on semantic similarity measurement and provides a definition for information retrieval.

2.1 Semantic Similarity

Similarity, the degree to which entities, concepts, or scenes resemble one another, is a foundational topic in many areas of cognitive science [1]. Semantic similarity measurement refers to the process of calculating an interval scaled value of

the proximity of the meanings. The importance of similarity measurement for categorization has been demonstrated through the observation of prototype effects, which show that objects are classified based on their semantic distance to an idealized prototype [18]. In GIScience semantic similarity has been an increasingly important topic, especially with respect to geographic information retrieval and the geospatial semantic Web [10]. Context awareness is an important (though often overlooked) component to any cognitively plausible similarity theory [11,16,17]. One approach to identify saliency weights for a given context was introduced by Tversky [19]. This approach uses the notion of *diagnosticity*, which indicates that the entity set being compared has a diagnostic effect of making certain features more salient with respect to their similarity. Rodriguez and Egenhofer [11] introduced the Matching-Distance Similarity Measure (MDSM) for measuring the similarity of geospatial features represented in a feature-based ontology. MDSM incorporates diagnosticity by utilizing the variability and commonalities of features in the ontology to determine their salience weights. These weights are then used to compute similarity. The Sim-DL similarity server implements a context-sensitive measure for concepts specified in description logics used on the semantic Web [12]. SIM-DL automatically adjusts the similarity of relations and primitive concepts based on context parameters provided by the user. Raubal [7] formalizes context in a similarity measure for geometric conceptual space representations by applying weights to the individual dimensions.

2.2 Geographic Information Retrieval

While information retrieval is an interdisciplinary research field including work on indexing and data storage, we focus on the relevancy relationship used to judge whether discovered information matches the user's needs. Formally, as shown in equation 1, information retrieval is about the degree of relevance between an object or set of objects and information desired by the user. The information sought is specified not only by an explicit query but also by implicit and inferred information gleaned from the user (e.g., a formal representation of personalization variables) [20]. In the context of Geographic Information Retrieval (GIR) the implicit information is spatially context-sensitive [21]. For example, the location of the user gives a GIR application implicit information that can be used to refine or otherwise alter search results to a local area.

$$IR = m[\mathfrak{R}(O, (Q, \langle I, \mapsto \rangle))] \tag{1}$$

where

- $\mathfrak{R}$ is the relevance relationship,
- O is a set of objects,
- Q is the user's query,
- I is implicit information,
- $\mapsto$ is inferred information,
- and m is the degree (or certainty) of relevance.

For the purpose of this work we use similarity as relevance relationship, while the query and compared-to objects are concepts from geo-ontologies.

3 Reification and Similarity

There are at least five major approaches to semantic similarity measurement, those based on computing feature overlap, on counting transformation steps, finding alignments, computing graph-distance in a network, and those based on geometric spaces; see [1,22] for recent overviews. As each of these approaches has its benefits and drawbacks, most modern similarity theories combine them to increase expressivity. For instance, similarity theories based on features or geometry are limited in their ability to handle relationships and are therefore enriched by network-based measures to form a hybrid model [11,23,12]. While there has been some work on translating and combining feature-based and geometric approaches in cognitive science [24,25], this topic has not received much attention in ontology engineering so far – a notable exception being work on ontological design patterns [26]. A classic translation example is the representation of dimensions such as length by sets of nested features [25].

From an ontological perspective, and as recently proposed by Scheider et al. [27], we argue that features are *fictions* which result from reifications[1] of (directly) perceivable observations. This shifts the debate from computational aspects to questions of granularity. Just like with the representation of geographic features, such as cities or transportation infrastructures, changes happen from points and polylines to polygons with scale, the description of concepts changes from features to regions in geometric space. This is the same process as applied in creating feature hierarchies or complex dimensions[2]. An ontology of land-use may list `Afforested` as feature type, while on a more detailed level the same notion can be modeled as minimum percentage value on the `CrownCover` dimension.

From the perspective of similarity measurement, we can regard feature-based similarities as coarser grained versions of similarities computed by geometric approaches. Hence, we can switch between them depending on the required granularity (as long as we can re-reificate the features). Note, however, that as feature-based similarity computes overlap while geometric approaches compute distance in a vector space the semantics of similarity changes with the translations.

The role context plays with respect to the relation between similarity and classification differs depending on whether a feature-based or geometric representation is used (figure 1). Diagnosticity in the feature-based representation assumes an *a priori* classification to determine weights on features that in turn are used as inputs for measuring similarity. Hence, classification generates similarity. The geometric approach, in contrast, is the opposite – classification is the result of a distance-based similarity function where context is represented by saliency weights on the dimensions. Though the dimensions used to measure similarity may be chosen *a priori*, the regions that represent the classes are not

[1] We restrict the notion of reification to the objectification of relations or dimensions.
[2] This is similar to the shift from prototypes modeled as points in a geometric space towards regions. However, this involves slightly changing semantics.

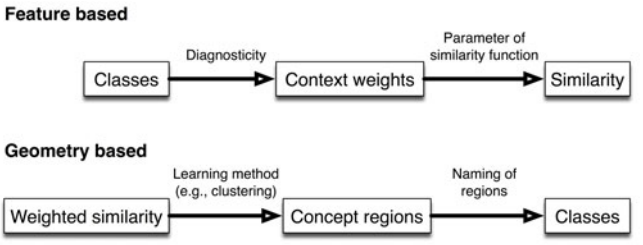


Fig. 1. Relationship of similarity and classification in different representations

(see also [28]). Thus, the definition of classes depends on the context. These differences have important implications for switching between granularities. If we consider features as analogous to regions in the geometric representation then reification can be thought of as the *labeling* of a particular classification (for a particular context) in the geometric representation.

4 Semantic Referencing

Semantic similarity measures are especially beneficial for navigating and browsing through large knowledge bases, i.e., for information retrieval, as well as for ontology engineering. They can be used to reduce the burden of understanding formal definitions [29], are more flexible than rigid (keyword) matching approaches, and help establishing new relationships between information. All these use cases, however, require that the similarity measures are cognitively plausible, i.e., that their rankings correlate with those from human users. As similarity is highly context sensitive, most recent similarity theories implement various context models [16]. While (semi) automatic weights are applied in many cases, they can only roughly approximate the similarity drift caused by additional information which is not explicitly stated in the user's query; see section 2.2.

A promising approach would be to combine weights with additional user feedback, i.e., allow users to influence the weighting process. However, assigning weights for features or dimensions of large ontologies would be a time consuming and error-prone task. To have perfect information about the user's preferences would require manual weights for all features and dimensions; taking asymmetry into account would even double the number of required pairwise comparisons. Finally, a user would have to take abstract decisions such as weighting the similarity between `Afforested` and `Artificial` which both may be features in a forestry ontology. Consequently, a feasible solution has to infer weights from partial information. In principle, there is an infinite number of possible context weights and their combination, which shifts the problem to an optimization challenge related to classical work from multi-criteria analysis.

In previous work, we have shown how users and domain experts can compare their own similarity estimations to the rankings produced by a similarity server to estimate whether the investigated ontology fits their purpose [29];

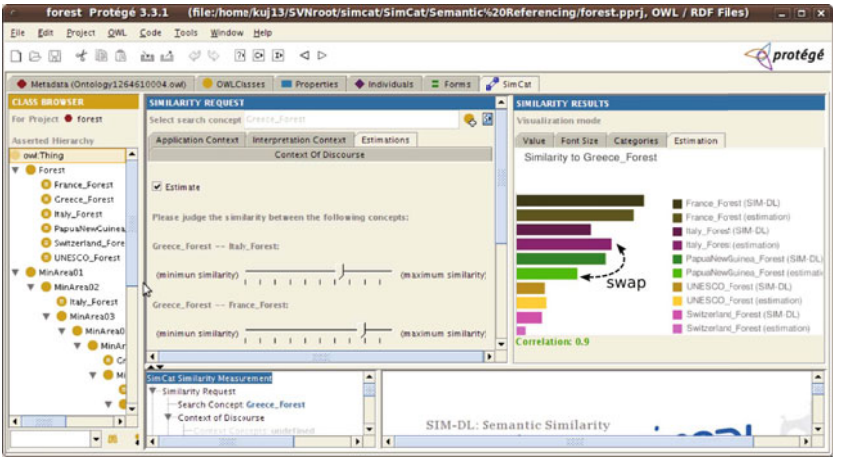


Fig. 2. Comparing the SIM-DL similarity estimations with those made by the user. Users should be able to swap ranks and hence influence weights.

see figure 2. So far, this approach had two shortcomings that could not be resolved. First, the estimations were done by computing rank-correlations (or concordance and rank-correlation in case multiple users were involved) which are not necessarily cognitively plausible. For example, the relative position in the ranking was not taken into account. Second, the system could only tell the domain experts whether the ontology potentially reflects their views or not, but did not offer a way to adjust the similarity weights produced by the similarity reasoner.

In this work, we propose a method to overcome both shortcomings. First by replacing Spearman's rank correlation coefficient with the *DIR* measure, and second by allowing the users to swap ranking positions to adjust weights semi-automatically; see *swap* in figure 2. DIR is a cognitively plausible dissimilarity measure for information retrieval result sets. It is based exclusively on result rankings and therefore applicable independent of the retrieval method. Unlike statistical correlation measures, DIR reflects how users quantify the changes in information retrieval result rankings [17]. It is defined as a symmetric function, which calculates the shift every concept undergoes when a query is posed in different contexts. A weighting function insures that shifts at the top of the rankings are emphasized. Note that we do not present abstract features or dimensions to the user but selected concepts ranked by their similarity. Moreover, the user does not need to take pair-wise decisions but directly changes the position of target concepts in the similarity ranking. These changes are then used to adjust the feature or dimension weights.

In analogy to georeferencing we call the process in which the weights get recomputed based on partial information provided by the user the *semantic referencing of similarities*. Georeferencing is the act of identifying a direct or indirect relation between an entity and a geographical position in space. In photogrammetry,

control points on the ground are used to fix the scale of the photographs. This can be simply done by measuring the natural distance between two points on the ground that can also be identified on the photographs. If a high degree of accuracy is required, then premarked points on the ground rather than natural features are used and based on the ground and picture coordinates a transformation is calculated [30]. While it is useful to think of the user adjustments as a kind of known control similarities (adequate to the user's conceptualization), there are also clear differences between both methodologies. These are grounded in the fact that semantic referencing has to cope with an arbitrary number of dimensions, not all of them can be adjusted by swaps in a single ranking, and that human notions of distance do not necessarily fulfill the metric requirements.

In the following we introduce a basic algorithm schema and diagnosticity functions for feature-based approaches. To demonstrate that our approach is generalizable we discuss which extensions are necessary for geometric models. We also introduce variability and commonality measures for geometric similarity measures which have not been investigated so far.

4.1 Semantic Referencing for Feature-Based Similarity Measures

For reasons of simplification, and in accordance with the classical feature-based theories, we assume concepts are defined by the intersection of more primitive ones which in turn can be further decomposed into features. Consequently, we leave logical negation, disjunction, and relationships between concepts and individuals aside. In such a representation language role-filler pairs can be represented as single features such as `NextToTransportationInfrastructure` which, as argued above, are reifications. For lack of space, we reduce the feature-based similarity between concepts to a ratio of common versus distinct features leaving asymmetry aside. Moreover, we only discuss *commonality* as diagnosticity measure and leave *variability* aside. The notion of asymmetry in Tversky's contrast model and variability have been extensively discussed in the literature [11]. Both can be included in the presented algorithm without major modifications, e.g., variability is just the inverse of commonality.

We assume that a user defines a query Q, in our case by selecting a search concept C_s, using a graphical user interface; for instance the SIM-DL Protégé plug-in or the semantics-based gazetteer web interface [16,8]. Instead of setting fixed weights, the task is to infer ($\mapsto$; see section 2.2) the weights from (explicit and implicit) information (I) provided by the user. We further assume that O, the set of objects in the information retrieval definition, is a set of target concepts $c_{t1}, ..., c_{tn}$ from the examined ontology. Different solutions have been proposed to determine which concepts should serve as search and target concepts. The Literature about multi-criteria analysis with partial information proposes to take examples in which the users are experts. Others propose to use the concepts which have been mentioned and grouped together most often during ontology engineering and knowledge acquisition tasks [29]. For populated ontologies, those concepts with the highest count of individuals may be a good choice as they have the highest probability to be used subsequently.

Listing 1.1. Basic Algorithm; one swap per turn version

```
1  for (feature : ⋃ C_{t_i} ∪ C_s)
2    computeGlobalDiagnosticity(              // See equations 3, 10.
3      computeLocalDiagnosticity(feature)); // See equations 2, 6, 8, 9.
4
5  for (C_t : O)
6    sortByDescendingDiagnosticity(C_t);
7    computeSimilarity(C_s, C_t);
8
9  theoryRanking = sortByDescendingSimilarity(O);          ⟶ [USER]
10 humanRanking = retrieveRanking();                       ⟵ [USER]
11
12 if (computeDIR(theoryRanking, humanRanking) ≤ threshold)
13   terminate; // No weight adjustment required.
14
15 candidateFeatureList(O); // See equation 4 and section 4.2.
16 modifyDiagnosticities(
17   humanRanking.swapFrom, humanRanking.swapTo); // See equations 5, 11.
```

Listing 1.1 shows the main steps to readjust the weights according to the ranking changes proposed by the user. First, the diagnosticity of each feature (f) has to be computed and normalized: see equations 2 and 3.

$$localDiagnosticity(f) = \frac{|\{f|f \in \bigcup C_{t_i} \cup C_s\}|}{\{|\bigcup C_{t_i} \cup C_s|\}} \tag{2}$$

$$globalDiagnosticity(localDiagnosticity) = \frac{localDiagnosticity}{\sum_{f_1}^{f_n} localDiagnosticity(f_i)} \tag{3}$$

Next, the features for each concept definition are ordered by their diagnosticity and the similarity for all search concept - target concept pairs is computed using a feature-based (or geometric) theory. As shown in lines 8 and 9, the resulting similarity ranking is presented to the user who can decide to swap two positions in the ranking (per turn). The user interface has to give the user the possibility to actively move a concept up or down as the choice of directions matters, i.e., the changes are asymmetric. Next, the DIR measure is used to determine how dissimilar both rankings are and whether an adjustment is necessary.

$$candidateFeatureList(O) = \{f|(f \in C_s) \wedge ((f \in C_{sf}) \bigoplus (f \in C_{st}))\} \tag{4}$$

As indicated in equation 4, only those features are candidates for weight adjustment which appear in the source and one of the target concepts (but not both). This is not required for theories which support similarity *between* features/dimensions such as SIM-DL or geometric approaches which will be discussed below.

$$\begin{aligned} MOD \;:\; & (x \pm C_{sf}[0].globalDiagnosticity) * sim(C_s[y], C_{sf}[0]) \\ & + (\frac{x}{2^1} \pm C_{sf}[1].globalDiagnosticity) * sim(C_s[y], C_{sf}[1]) \\ & + (\frac{x}{2^2} \pm C_{sf}[2].globalDiagnosticity) * sim(C_s[y], C_{sf}[2]) + ... \frac{x}{2^n} \\ & = sim(C_s, C_{st}) + 0.01; \end{aligned} \tag{5}$$

Equation 5 shows how the weights are increased (+) or decreased (-) based on the user's modifications where C_{sf} is the swapped-from and C_{st} the swapped-to concept in the ranking. y is the to-be-compared feature in the C_s list. Instead of arbitrary changes to multiple weights we use a power function to model the Max Effect described in cognitive science studies [31]. This effect describes the tendency to favor a particular reasoning strategy which has turned out to be successful in previous similarity estimations and could be compared to the Matthew Effect in social science. Features which are more diagnostic gain even more diagnosticity while the diagnosticity of others increases slower. So far our algorithm considers local optimizations. It tries to ensure that the weights reflect the swap in the first place and put less emphasis on other parts of the ranking (which can still be adjusted in the next turn). Note that modifying the weights does not always successfully change the ranking or guarantees that the process converges at all. This is especially the case if the user's initial conceptualization differs clearly from the computational representation in the ontology[3] or if users take irrational decisions. In this case the ontology is unsuitable for the given context. After the adjustment, the new diagnosticity weights are used within the ontology to better approximate the user's preferences.

4.2 Measuring Diagnosticity in Geometric Representations

In this section we extend the MDSM measures of commonality and variability to geometry-based representations.

Case 1 - Commonality. We have a set of concepts that are represented as incomplete vectors (i.e., points) in a continuous multidimensional space. That is, for any given concept values may be undefined for one or more of the dimensions. In this case we are interested in determining the diagnosticity of each dimension based on how many concepts have a value defined for it. Let $C = \{c_1, c_2, \ldots, c_n\}$ be the set of concepts. The probability $p(c, d)$ that a concept $c \in C$ has a value defined for a given dimension d is equal to the number of concepts defined for d over the total number of concepts in C. The *commonality* diagnosticity (CD) of a dimension d is thus defined in equation 6.

$$CD = \frac{p(c, d)}{\sum_{i=1}^{m} p(c, d_i)} \tag{6}$$

Case 2 - Variability. We have a set of concepts that are represented as complete vectors in a continuous multidimensional space. In this case, the concepts are defined using the same dimensions, but they still differ semantically in that they are represented by different points in the space. This situation will occur when representing a set of observations construed as exemplars of a concept (or

[3] The user does not need to know or understand the formal definitions of concepts displayed in the ranking which at the same time is a major benefit of our approach as discussed before.

in ontological language an enumeration of individuals) and that are measured using the same methods.

Since there are no differences between the dimensions, diagnosticity is measured in terms of deviation of the data along the different dimensions. The semantic interpretation of the deviation depends on whether the data points represent different exemplars of the same concept (e.g., different definitions of forest) or in fact different concepts (e.g., forest, woodland, chaparral, etc.). Our hypothesis is that if a set of instances of the same concept varies little along one dimension (x) and a lot along another dimension (y) then dimension x is more salient and therefore more diagnostic. In the case of different concepts the opposite is true. Intuitively, we want to identify which quality values are most alike for the instances of the same concept and which ones help us to distinguish between different concepts.

The proposed method is to compare the mean absolute deviation (MAD) of the data values (X) for each dimension. To maintain consistency with MDSM terminology we call the MAD value of a dimension its *variability*.

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|x_i - m(X)| \tag{7}$$

The data must first be normalized to [0,1] along each dimension so that the MADs can be compared. Depending on the data, different normalization techniques may be necessary. In general, a Min-max normalization will be sufficient, though in the case that the data have a fixed range then the range minimum and maximum is preferred (e.g., any ratio scaled dimension will have a minimum value of zero). It is noted, however, that Min-max normalization is sensitive to outliers. Equation 8 shows the *variability* diagnosticity (VD) of a dimension for the case when different instances of the same concept are represented.

$$VD_{same} = \frac{1 - \frac{MAD(d)}{\sum_{i=1}^{m} MAD(d_i)}}{m-1} \tag{8}$$

VD in the case of different concepts is defined in equation 9.

$$VD_{diff} = \frac{MAD(d)}{\sum_{i=1}^{m} MAD(d_i)} \tag{9}$$

Combining Commonality and Variability. The scenario presented in case 1 will always include different distributions of values along each of the dimensions as well, so we define a diagnosticity measure for a given dimension d (equation 10) that combines the two measures listed above. However, the VD measure (see equations 8 and 9) is changed slightly to ignore any undefined values when calculating the MAD.

$$diagnosticity(d) = \frac{CD(d) \times VD(d)}{\sum_{i=1}^{m} CD(d_i) \times VD(d_i)} \tag{10}$$

Modifying Diagnosticities for Geometric Representation. Equation 11 shows an extended MOD function for calculating new diagnosticities for dimensions in geometry based representations.

$$\begin{aligned} MOD \ : \ & (x + diag(dimP[0])) * sim(C_s[dimP[0]], C_{sf}[dimP[0]]) \\ & + (\frac{x}{2^1} + diag(dimP[1])) * sim(C_s[dimP[1]], C_{sf}[dimP[1]] + ... \\ & + (x - diag(dimM[0])) * sim(C_s[dimM[0]], C_{sf}[dimM[0]]) \\ & + (\frac{x}{2^1} - diag(dimM[1])) * sim(C_s[dimM[1]], C_{sf}[dimM[1]] + ... \\ & = sim(C_s, C_{st}) + 0.01; \end{aligned} \tag{11}$$

Contrary to the feature method (see equation 4), the candidate dimension list includes all dimensions D shared by C_s, C_{sf}, and C_{st}, because different concepts may share the same dimensions but vary in terms of the values along those dimensions. The diagnosticities of each dimension can be calculated by any of the methods described above, depending on the application. It compares the similarities of C_{sf} and C_{st} to C_s, and constructs two sorted queues, $dimP$ and $dimM$ based on which of the two target concepts is most similar to the search concept for each given dimension (see listing 1.2). The queues are sorted by how large the difference is between the similarities. These queues are used to generate positive and negative power functions, which are combined and solved to generate new diagnosticities for each of the dimensions.

Listing 1.2. Determining which dimensions increase/decrease diagnosticity

```
1  for (d : D) // for each dimension
2    fromToSimDiff = sim(C_s[d], C_sf[d]) - sim(C_s[d], C_st[d])
3    if (fromToSimDiff > 0) // similarity of C_s to C_sf < C_s to C_st
4      dimP << d, fromToSimDiff  // add to + queue sorted by difference
5    else
6      dimM << d, fromToSimDiff // add to - queue sorted by difference
```

5 Application

Data about forest cover in a country are dependent in part on the definition (i.e., semantics) of *forest* used by that country. In order to compare these data across different countries it is helpful to identify which definitions are more similar to one another. This similarity information can then be used to evaluate the degree to which the forest cover data from different countries are comparable. In this section, we present the usage of the semantic referencing algorithm to semi-automatically calculate the similarity of different *forest* definitions. For this example we use a geometry-based representation, where each forest definition is represented as a point in a three dimensional space. The three dimensions are minimum area, minimum crown height, and minimum tree height[4].

[4] Source: http://www.affrc.go.jp/satellite/shokusei/EOSD/Background/Gyde_Lund_Definitions_of_Forest_RAD/DEFpaper.html

Table 1. Sample forest definition control points and diagnosticities of dimensions

	Min area (ha)	**Min crown cover (%)**	**Min tree height (m)**
France	2.0	10.0	
Greece	0.5	10.0	
Italy	0.2	20.0	
Papua New Guinea	100.0	10.0	5.0
Switzerland		20.0	
UNESCO		40.0	5.0
Commonality (CD)	0.333	0.5	0.167
Variability (VD_{same})	0.092	0.408	0.5
Diagnosticity	**0.096**	**0.642**	**0.262**

5.1 Calculating Diagnosticity

The user first selects control points, which are used to calculate the diagnosticity of each of the dimensions. For this particular example we assume some domain expertise on the part of the user regarding forest definitions, so that they have an internal conceptualization with which they can compare the rankings. Table 1 shows a sample selection of control points along with commonality (CD), variability (VD_{same}), and the combined $diagnosticity(d)$ calculations based on 6 sample control points. The minimum crown cover dimension is the most diagnostic as it shows a low variability (which is a high diagnostic indicator when comparing exemplars of the same concept) and high commonality as it is defined for all sample countries. The abstract semantic referencing algorithm can use any of the three diagnosticity measures to identify which dimension is most salient; in this case we use the combined measure.

5.2 Iterating through the Semantic Referencing Algorithm

For this geometry-based representation we assume a semantic distance (dissimilarity) between two forest definitions is equal to the weighted Manhattan distance between the two vector representations where the weights are the diagnosticities. Similarity is simply defined as 1 - *semantic distance*. If a dimension is defined for the source forest but not in the target then the distance is considered maximal (i.e., 1 on a dimension normalized to [0..1]). If the dimension is undefined for the source the distance along that dimension is considered to equal zero. Using this measure we calculate a ranking of the target forest definitions to the source target, which in our example is the definition of Greek forest with the following similarity ranking: 1. France; 2. Italy; 3. Papua New Guinea; 4. Switzerland; 5. UNESCO. The user can choose to accept the ranking or adjust it by moving a target up or down. Say the user wants to move Papua New Guinea to Italy's ranking. Using the geometry-based MOD function (see equation 11) the diagnosticity of minimum area is reduced and the diagnosticity of minimum crown cover is increased, because Papua New Guinea's and France's forest definitions are more similar along the minimum crown cover dimension than Italy's and

France's are (and vice versa for minimum area). Similarity rankings are recalculated using the new diagnosticity values and the above process is reiterated until the user gets an acceptable ranking, which is then used for similarity measures on a wider set of forest definitions in a traditional information retrieval setting.

6 Conclusions and Future Work

In this paper we have discussed the relationship between feature-based and geometric similarity theories from the viewpoint of ontology engineering, defined diagnosticity for geometric similarity measures, and provided a novel methodology for adjusting weights based on user preferences. While the user provides an explicit query, the main innovation in the semantic referencing approach is to use implicit information. Instead of presenting abstract feature or dimension pairs, we propose to let the users adjust precomputed inter-concept similarity rankings to learn about their contextual preferences and apply the extracted weights to the ontology.

While this is the first step, our long-term vision is to apply the semantic referencing methodology automatically based on the user's similarity estimations as depicted in figure 2. Such approach would be more intuitive and could be directly integrated into our similarity servers. This, however, would require more sophisticated and global optimization functions. The challenge in developing such functions is not to find *a* working algorithm, but to ensure its cognitive plausibility. It turns out that weighting approaches have their limitations and may not be able to model the user's needs in all cases. The presented work is based on established and well tested measures, complex optimization approaches may, however, require changes in the alignment process or flexible distance metrics differing among dimensions, and hence, will also require extensive human participants test. Moreover, the user's context can also contain additional implicit information to further refining the results. For example, certain characteristics of geographic feature types may be more or less salient depending on the country (and language) of origin of the search. The visualization and interaction with conceptual spaces is also an important field for further research – we believe that parallel coordinate plots may be an interesting solution to some of these challenges.

Finally, there are several improvements to make the diagnosticity measures more robust. First, within the geometric representation there is an assumption that the dimensions are orthogonal allowing us to measure the variability of each dimension independently, but this is not necessarily true if there are correlations between these dimensions. Second, the measure of a domain's variability based on a set of property regions in the geometric representation is simplified to a point measure, which does not consider the structure and size of the regions. One parameter to consider is the degree and kind of overlap among different regions. Other measures of diagnosticity such as the information entropy of a set of data values should be explored and evaluated as well. A region connection calculus can be used to reify the topology of the regions to feature-based properties and their relations, thus suggesting a strategy for developing a diagnosticity measure that encompasses hybrid feature-geometry representations [32].

Acknowledgments

This work is supported by a UCSB faculty research grant and NGA-NURI grant HM1582-10-1-0007.

References

1. Goldstone, R.L., Son, J.: Similarity. In: Holyoak, K., Morrison, R. (eds.) Cambridge Handbook of Thinking and Reasoning, pp. 13–36. Cambridge University Press, Cambridge (2005)
2. Rissland, E.L.: Ai and similarity. IEEE Intelligent Systems 21(3), 39–49 (2006)
3. Hofstadter, D.R.: Gödel, Escher, Bach: An Eternal Golden Braid. Basic Books, New York (1999)
4. Gärdenfors, P.: Conceptual Spaces - The Geometry of Thought. Bradford Books. MIT Press, Cambridge (2000)
5. Nedas, K., Egenhofer, M.: Spatial similarity queries with logical operators. In: Hadzilacos, T., Manolopoulos, Y., Roddick, J., Theodoridis, Y. (eds.) SSTD 2003. LNCS, vol. 2750, pp. 430–448. Springer, Heidelberg (2003)
6. Li, B., Fonseca, F.: Tdd - a comprehensive model for qualitative spatial similarity assessment. Spatial Cognition and Computation 6(1), 31–62 (2006)
7. Raubal, M.: Formalizing conceptual spaces. In: Varzi, A., Vieu, L. (eds.) Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004), Torino, Italy, November 2004. Frontiers in Artificial Intelligence and Applications, vol. 114, pp. 153–164. IOS Press, Amsterdam (2004)
8. Janowicz, K., Schwarz, M., Wilkes, M.: Implementation and evaluation of a semantics-based user interface for web gazetteers. In: Workshop on Visual Interfaces to the Social and the Semantic Web, VISSW 2009 (2009)
9. Ahlqvist, O.: Extending post classification change detection using semantic similarity metrics to overcome class heterogeneity: a study of 1992 and 2001 national land cover database changes. Remote Sensing of Environment 112(3), 1226–1241 (2008)
10. Egenhofer, M.: Toward the semantic geospatial web. In: GIS 2002: Proceedings of the 10th ACM international symposium on Advances in geographic information systems, pp. 1–4. ACM, New York (2002)
11. Rodríguez, A., Egenhofer, M.: Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. International Journal of Geographical Information Science 18(3), 229–256 (2004)
12. Janowicz, K., Wilkes, M.: $SIM - DL_A$: A Novel Semantic Similarity Measure for Description Logics Reducing Inter-concept to Inter-instance Similarity. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 353–367. Springer, Heidelberg (2009)
13. Cruz, I., Sunna, W.: Structural alignment methods with applications to geospatial ontologies. Transactions in GIS 12(6), 683–711 (2008)
14. Adams, B., Raubal, M.: A metric conceptual space algebra. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 51–68. Springer, Heidelberg (2009)
15. Goldstone, R.L., Medin, D.L., Halberstadt, J.: Similarity in context. Memory & Cognition 25, 237–255 (1997)

16. Janowicz, K.: Kinds of contexts and their impact on semantic similarity measurement. In: 5th IEEE Workshop on Context Modeling and Reasoning (CoMoRea 2008) at the 6th IEEE International Conference on Pervasive Computing and Communication (PerCom 2008), pp. 441–446 (2008)
17. Keßler, C.: What's the difference? - a cognitive dissimilarity measure for information retrieval result sets. Knowledge and Information Systems (forthcoming)
18. Goldstone, R.L.: The role of similarity in categorization: providing a groundwork. Cognition 52(2), 125–157 (1994)
19. Tversky, A.: Features of similarity. Psychological Review 84(4), 327–352 (1977)
20. Dominich, S.: The Modern Algebra of Information Retrieval. Springer, Heidelberg (2008)
21. Keßler, C., Raubal, M., Wosniok, C.: Semantic rules for context-aware geographical information retrieval. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 77–92. Springer, Heidelberg (2009)
22. Schwering, A.: Approaches to semantic similarity measurement for geo-spatial data - a survey. Transactions in GIS 12(1), 5–29 (2008)
23. Schwering, A., Raubal, M.: Spatial relations for semantic similarity measurement. In: Akoka, J., Liddle, S.W., Song, I.-Y., Bertolotto, M., Comyn-Wattiau, I., van den Heuvel, W.-J., Kolp, M., Trujillo, J., Kop, C., Mayr, H.C. (eds.) ER Workshops 2005. LNCS, vol. 3770, pp. 259–269. Springer, Heidelberg (2005)
24. Navarro, D., Lee, M.: Combining dimensions and features in similarity-based representations. In: Becker, S., Obermayer, S.T.,, K. (eds.) Advances in Neural Information Processing Systems, vol. 15, pp. 59–66. MIT Press, Cambridge (2003)
25. Gati, I., Tversky, A.: Representations of qualitative and quantitative dimensions. Journal of Experimental Psychology: Human Perception and Performance 8(2), 325–340 (1982)
26. Gangemi, A.: Ontology design patterns for semantic web content. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 262–276. Springer, Heidelberg (2005)
27. Scheider, S., Probst, F., Janowicz, K.: Constructing bodies and their qualities from observations. In: 6th International Conference on Formal Ontology in Information Systems (FOIS 2010 forthcoming)
28. Ahlqvist, O.: In search for classification that support the dynamics of science? the fao land cover classification system and proposed modifications. Environment and Planning B: Planning and Design 35(1), 169–186 (2008)
29. Janowicz, K., Maué, P., Wilkes, M., Braun, M., Schade, S., Dupke, S., Kuhn, W.: Similarity as a quality indicator in ontology engineering. In: Eschenbach, C., Grüninger, M. (eds.) 5th International Conference on Formal Ontology in Information Systems, October 2008, vol. 183, pp. 92–105. IOS Press, Amsterdam (2008)
30. Kraus, K.: Photogrammetry: Geometry from Images and Laser Scans, 2nd edn. Walter de Gruyter, Berlin (2007)
31. Medin, D., Goldstone, R., Gentner, D.: Respects for similarity. Psychological Review 100(2), 254–278 (1993)
32. Gärdenfors, P., Williams, M.A.: Reasoning about categories in conceptual spaces. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001), pp. 385–392 (2001)

User-Centric Time-Distance Representation of Road Networks

Christian Kaiser[1,2], Fergal Walsh[1],
Carson J. Q. Farmer[1], and Alexei Pozdnoukhov[1,⋆]

[1] National Centre for Geocomputation, National University of Ireland, Maynooth
[2] Institute of Geography, University of Lausanne, Switzerland
alexei.pozdnoukhov@nuim.ie

Abstract. This paper presents a new algorithm for computing time-distance transformations of a road network based on modified multi-dimensional scaling. The algorithm is designed to perform on a real-world road network, and provides alternative visualisations for travel time cognition and route planning. Several extensions are explored, including user-centric and route-centric road map transformations. Our implementation of the algorithm can be applied to any locality where travel time road network data is available. Here, it is illustrated on road network data for a rural region in Ireland. Limitations of the proposed algorithm are examined, and potential solutions are discussed.

Keywords: Road networks, time-distance cartograms, multi-dimensional scaling.

1 Introduction

With the increasing availability of navigation systems and interactive maps, spatial information is becoming increasingly accessible to the average user. Geographical maps have the potential to display large amounts of spatial information which can lead to 'information overload', unless an emphasis is placed on intelligent and adaptive presentation of information. Traditionally, to increase the amount of information contained within a map, additional contextual layers are required. The display of time information however, requires special consideration.

One alternative visualisation of spatial data comes from the use of time-distance mapping. In this context, a time-distance map is based on the travel time on a network (e.g. road network), where the space is deformed such that the length of the links (roads) between the nodes (intersections) of the network are proportional to their travel time. In this representation, the normal geographical scale (Euclidean distance) of the network can be replaced by a *time*-distance scale, using hours instead of kilometres as the unit of measurement. An interesting example of this type of representation is the time-distance transform of the London Tube Map with a static [1] and dynamic version [2]. These types of

⋆ Corresponding author.

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 85–99, 2010.

artist-designed representations of time-distance currently appear to be the most graphically appealing maps; the average user may potentially find them more useful, as the deformation does not strongly interfere with the map content. The classic London Underground Map has been designed for readability and is already an approximation of a time-distance map, such that stations are at equal distances from each other, reflecting the fact that metro travel time is essentially independent of the real-world distance between stations [3].

Some work has been done in the past concerning automatic algorithms for displaying time-distance information [4,5,6]. While effective for geographic information professionals, these representations do not appear to be sufficiently intuitive to aid the average user in everyday decision making, as suggested by the fact that their use in maps for lay-people is uncommon. Conversely, cartograms [7,8,9] have been successful at representing areal data, giving promise to a similar representation of time-distance. To date, the attempts to integrate time with a geographical view of space either were aimed at professionals, or required considerable manual design intervention.

In this paper, we propose an automated algorithm designed to enhance the interpretation of road maps through the integration of the time dimension. Our algorithm is designed to maintain some characteristics of conventional geographical road maps, such as the orientation of the road segments and the location of selected landmarks. These constraints provide a time-distance representation which is sufficiently realistic, making it possible to be used as a complement or replacement for a traditional road map.

We propose two different implementations of the algorithm: (1) a user-centric mode in which the user is interested in features or locations which are in close proximity to their current location, and (2) a route-centric representation, where the user is interested in map features which are proximal to a specified route along the road network. Further, we provide a simple solution to the out-of-sample problem which provides a means of transforming satellite imagery, which may be used to provide context to the embedded road map visualisation. Our implementation of the algorithm is based on the OpenStreetMap (OSM) data (`www.osm.org`) and can be applied anywhere where travel time data for road networks is available.

2 Previous Work

Time-space maps have had a long history in geography with works dating back to the 1960s [10,11]. Most time-distance maps are used to represent the distances between all nodes in a network as accurately as possible, and in order to achieve this, Multi-Dimensional Scaling (MDS) is commonly used [12,13,5]. MDS is designed to find the "best" configuration of network points by minimising a global error criteria [14]. However, MDS is not well suited for large networks with a high number of connections. In this case, the distortion becomes important and the visual representation can be quite confusing. Recently, Shimizu and Inoue [6] have created partial network time-space maps using a modified MDS approach,

where only a limited number of nodal distances are considered, for example the distances on the main transportation network. Additionally, they limit the deformation of the time-distance map by including the orientation of the network links in the constraints.

Denain and Langlois [15] have studied the utility of distance cartograms. In their opinion, isochrones are preferred because they are able to directly demonstrate time value-distances on the map without any geographical distortion. According to Denain and Langlois, the intent of making cartograms is to visualise a given phenomena, for example the reduction of the travel time due to the introduction of a high speed train network, without any link to geographic reality. But again, the authors focus on time-distance maps where all nodes on the network are involved.

User-centric representations of space have been successful in the past by integrating a local view into a global view, where the local view can be simply a zoomed view into the central region or be a somehow deformed map, e.g. a "fish-eye". This type of method focuses on exploration of the map at two zoom levels: (1) a global view (context view) outside of the centre, and (2) a more detailed local view (focus view) in the centre of the map. For example, Yamamoto, Ozeki and Takahashi [16] explore a fisheye approach for dynamic web map services using a central region of focus surrounding the user's location, along with a "glue region" connecting the two views. Similar studies combining a global and a local view have been conducted for network maps [17] or tourist maps [18].

We are not aware of any work that attempts to combine a topographic map together with a time-distance map with the purpose of enhancing the visual representation of a map from a user-centric perspective. In this paper, we present some possible approaches for transforming a traditional topographic or road map similar to those found in navigation devices and web-based maps, and provide a visualisation that may improve spatial cognition, and may also help to speed up decision making.

3 Methodology

MDS is a classic statistical technique which can be used to generate time-distance representations of spatial objects [14]. In Sec. 3.1, we review some of the relevant aspects of classic MDS in the context of time-distance mapping. In Sec. 3.2 to 3.6, modifications and extensions to classic MDS are introduced in order to improve the time-distance representation of real-world maps.

3.1 Classic Metric Multi-dimensional Scaling

The task of classical MDS scaling is to find M-dimensional coordinates for a given set of objects $\mathcal{V} = \{1, \cdots, n\}$ such that their pair-wise Euclidean distances equal the given dissimilarities $\mathbf{D} = \{d_{ij}\}$. In other words, the problem to be solved is:

$$\begin{aligned} &\text{Given} \quad \mathbf{D} = \{d_{ij}\}, \quad (i,j) = 1, \cdots, n, \\ &\text{find} \quad x_1, \cdots, x_n \in \mathbb{R}^M, \\ &\text{such that} \quad \|x_i - x_j\| = d_{ij} \quad \text{for all} \quad (i,j) = 1, \cdots, n. \end{aligned} \tag{1}$$

The solution is a configuration $\mathbf{X} = \{x_1, \ldots, x_n\}$ where each x_i is a point in the M-dimensional space corresponding to an object in $\mathcal{V}$. The complete explication to this problem, which has an exact solution if the dissimilarities d_{ij} are indeed Euclidean distances, was given by [19] using matrix algebra and eigenvalues analysis. Independently, an equivalent formulation some years later was described by [20]. An extension to the above method is possible if distances are known with errors, and is derived by minimization of a loss function operating with inner products. If one considers the dissimilarities as shortest path distances on a graph $(\mathcal{V},\mathcal{U})$ where $\mathcal{V}$ are the vertices and $\mathcal{U}$ the edges, the MDS is then a natural choice for graph drawing and network analysis [21], [22]. Classical scaling and its computational implementation is not, however, well suited for large scale problems, as it cannot handle missing values, or be adapted to favour a good fit to particular predefined pairs.

Stress minimization. As an alternative to classical scaling, one can consider the problem of directly minimising the misfit of the configuration to given dissimilarities [23]. This is done by finding a configuration $\mathbf{X} = \{x_1, \ldots, x_n\}$ such that we minimise the *stress* function

$$\sigma(\mathbf{X}) = \sum_{i<j} (d_{ij} - \|x_i - x_j\|^2). \tag{2}$$

While no method is known for constructing algebraic optimal solutions that minimizes (2), this optimization problem can be solved iteratively with a form of gradient descent. To bring some flexibility to the setting, the influence of particular pairs of data points can be adjusted by introducing weights $w_{ij} > 0$ as coefficients of the individual error terms:

$$\sigma_w(\mathbf{X}) = \sum_{i<j} w_{ij}(d_{ij} - \|x_i - x_j\|^2). \tag{3}$$

If the desired distance is not known, unreliable, or an exact fit is not required for a pair, this pair can be simply ignored by setting $w_{ij} = 0$. Different scaling of the weights, of the general form $w_{ij} = d_{ij}^{-p}$ can be considered, resulting for instance in Sammon's mapping [24] for $p = 1$ and elastic mapping [25] for $p = 2$.

A "stress majorization" technique was developed [26] to improve the direct minimization of (3). The stress function is substituted with a *convex* function which is greater than or equal to the stress (hence majorization), but easier to minimize algebraically and computationally as it has a unique global minimum. In practice, a simple and efficient iterative procedure [27] can be used to find a configuration $\mathbf{X}$:

$$x_i^{[t+1]} \leftarrow \frac{\sum_{j \neq i} w_{ij}(x_j^{[t]} + s_{ij}^{[t]}(x_i^{[t]} - x_j^{[t]}))}{\sum_{j \neq i} w_{ij}}. \tag{4}$$

where

$$s_{ij}^{[t]} = \begin{cases} \frac{d_{ij}}{\|x_i^{[t]} - x_j^{[t]}\|} & \text{if } \|x_i^{[t]} - x_j^{[t]}\| < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

This process is repeated until the relative decrease in stress is larger then a predefined threshold value $\epsilon > 0$.

Road network representations with MDS. Consider a road network as a weighted undirected graph $\mathcal{V}$,$\mathcal{U}$ of nodes with geographical coordinates $\mathcal{V} = \{x_1^{geo}, \ldots, x_n^{geo}\}$. Let $\mathbf{S}$ be a (sparse) matrix of distances between the adjacent nodes scaled according to the travel time. Define a shortest-path distance matrix on the graph computed from $\mathbf{S}$ as $\mathbf{G} = \{g_{ij}\}$.

To apply MDS to a road network, one must compute the configuration $\{x_1, \ldots, x_n\}$ for $\mathbf{D} = \mathbf{G}$ either by the classical method, or stress minimization. This produces a time-distance representation which attempts, on average, to reproduce all pairwise road distances. While these representations can be useful in particular applications, they suffer from a number of drawbacks. To motivate the following modifications to the MDS algorithms, consider an MDS applied to a particular road network, shown in Fig. 1. Note that in order to generate a more accurate representation of the travel times, additional nodes at regular intervals along the roads are added prior to the application of MDS. This also ensures that the most direct path in the time-distance map is the fastest one.

The global transformation shown in Fig. 1 (right) is difficult to interpret, due mostly to the fact that the transformation does not necessarily provide an intuitive representation of space. In other words, the map is overly warped, and distances and directions are difficult to interpret. Instead, we suggest a user-centric transform, which is aimed at (1) reproducing the travel distances

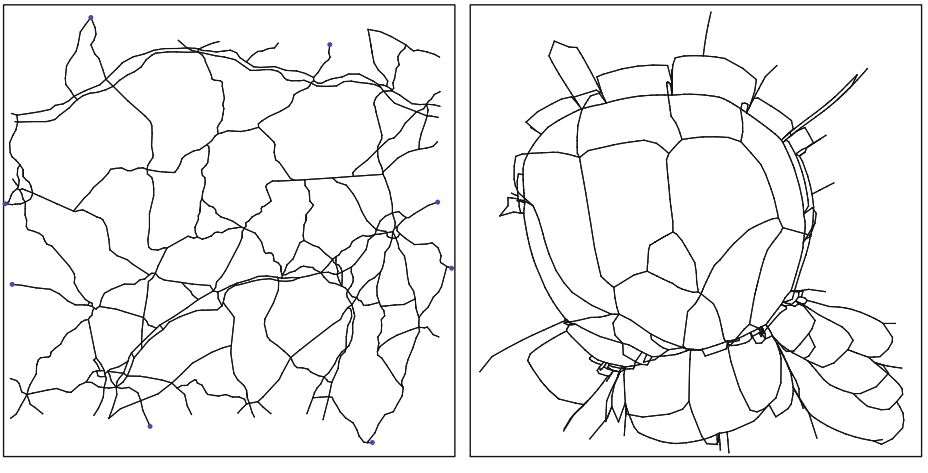


Fig. 1. Original road network (left) and its time-space representation obtained with classical MDS (right)

from a specific location (Sec. 3.2), (2) looking "similar" to a conventional map (Sec. 3.5), and (3) suited for graphical embedding in an untransformed map (Sec. 3.3). Using these extensions, we would like to produce navigation-oriented time-distance representations, which provides a desired time-distance alignment centred around a predefined path on a network (Sec. 3.4). Finally, we discuss the out-of-sample problem which has to be addressed to allow us to render other map geometries and warp satellite imagery of the area of interest (Sec. 3.6).

3.2 User-Centric Transforms

This extension allows the time-distance transformation to be centred around a given node x_c of the network. Let matrix $\mathbf{E} = \{e_{ij}\}$ be a matrix of Euclidean distances between the nodes, $e_{ij} = \|x_i^{geo} - x_j^{geo}\|$. Our aim is to transform road distances in the vicinity of x_c, while maintaining the geographical layout of the distant nodes. We approach this by introducing a weighting scheme to favour for shortest path road distances in the vicinity of x_c. First, by using a smoothed Heaviside step function we define a local region of radius $\rho > 0$ around x_c:

$$\lambda_{x_c} = \frac{1}{2}(1 + \tanh(\gamma(\rho\mathbf{I} - \mathbf{G}_c))), \tag{6}$$

where steepness of the step is controlled by the parameter $\gamma > 0$, $\mathbf{I}$ is a $(1 \times n)$ unit vector, and $\mathbf{G}_c$ is a column of $\mathbf{G}$ corresponding to x_c. The parameters $\rho > 0$ and $\gamma > 0$ allows one to adjust the geographical extent of the area to be transformed, and the smoothness of the transition from a non-transformed region into the time-distance representation of the vicinity of x_c. We then compute a weighted distance matrix $\mathbf{D}_c$ with elements

$$d_{ij} = \frac{\lambda_{x_c}^i + \lambda_{x_c}^j}{2} g_{ij} + (1 - \frac{\lambda_{x_c}^i + \lambda_{x_c}^j}{2}) e_{ij}, \tag{7}$$

where the superscripts i and j denote vector elements. The effective distance matrix $\mathbf{D}_c$ includes shortest path distances for all nodes in the vicinity ρ of x_c (shown with dark colours in Fig. 2 (left)) and preserves Euclidean distances between distant nodes (those in white colour in Fig. 2 (left)). It is then used for deriving a transformed network configuration with iterative optimization (4)-(5).

3.3 Preserving Landmark Locations

An exact match to pre-defined locations $\{x_k\}$ can be required for a subset of nodes $\{x_k^{geo}\}_{k=1,\ldots,K}$. It is particularly useful when one would like to keep the exact geographical location x_c^{geo} of the x_c and a set of boundary nodes in order to embed a transformed network into an untransformed road map. Enforcing the hard constraints $x_k = x_k^{geo}, \quad k = 1, \ldots, K$ would significantly complicate the optimization problem. However, a reasonable approximation can be achieved in a mean-square sense by adding a penalty $\sum_{k=1}^{K}(x_k - x_k^{geo})^2$ with some weight β to the stress function. This approach falls under the "soft constraints" category

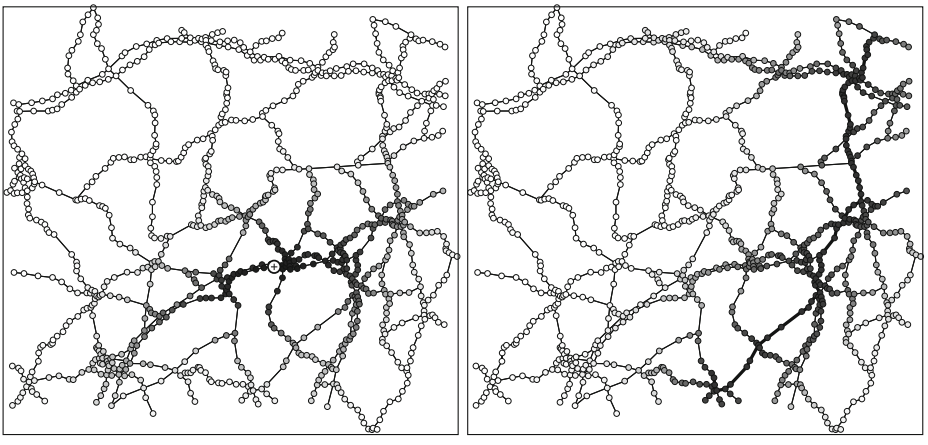


Fig. 2. Left: transformation vicinity for user-centric transform as defined by weights λ_{x_c}, Eq. (6). User location is marked with a crossed circle. Right: route-centric weight λ_r, Eq. (10). The selected route is shown in bold line. A white-to-black grey scale level corresponding to a $[0, 1]$-interval of the values of λ_{x_c}, λ_r is used in both graphs.

in [28]. Consequently, the optimization procedure (4) has to include the following update steps at every iteration:

$$x_k^{[t+1]} \leftarrow x_k^{[t]} - 2\beta(x_k^{[t]} - x_k^{geo}), \quad k = 1, \ldots, K. \tag{8}$$

An adaptive weighting scheme can be used to improve the fit of the distances between landmark nodes x_k by increasing the values of respective columns and rows of $\mathbf{W}$.

3.4 Route-Centric Transforms

Consider a predefined path a user may have chosen to follow on the road network. Suppose it consists of a sequence of adjacent nodes $\{x_r\}_{r=1,\ldots,R}$. In this case, we would like to produce a transform which preserves the time-distances between nodes along the selected route, while maintaining the pair-wise Euclidean distances between nodes not proximal to the route. Firstly, we compute the shortest path distances from all the nodes of the network to the selected route:

$$g_i^{path} = \min_{r=1,\ldots,R} g_{ir}, i = 1, \ldots, n. \tag{9}$$

We then introduce a weighting scheme for those nodes close to the route in analogy with Eq. 6:

$$\lambda_r = \left[\frac{1}{2}(1 + \tanh(\gamma(\rho\mathbf{I} - g_i^{path})))\right], i = 1, \ldots, n. \tag{10}$$

Here, λ_r is a vector containing weights for every node on the network describing its proximity to the selected route (refer to Fig. 2 for an illustration). The radius

of this region is controlled by user-defined parameter $\rho > 0$ and the steepness of the transition by $\gamma > 0$. These parameters define the geographical extent of the area to be warped into time-distances. This weighting scheme is used to compute the matrix of effective distances similarly to Eq. (7) for iterative stress minimization.

The next modification to the MDS algorithm allows us to warp the route and present it as a straight line on the graph, maintaining the map-like properties described above. Suppose the nodes on the shortest route include $\{x_r\}_{r=1,\ldots,R}$. This can be achieved by iterating the updates:

$$x_r{}^{[t+1]} \leftarrow x_r{}^{[t]} - 2\beta_{path}(x_r{}^{[t]} - x_r^{path}), \quad r = 1, \ldots, R, \tag{11}$$

where x_r^{path} are the desired locations of the nodes on the path, and β_{path} the stress function weight of the nodes. In the following applications, we align these nodes along a straight-line path connecting the origin (o) and destination (d) of the route, precisely reproducing the travel time distances along said route.

3.5 Preserving Angles

The MDS-related transform extensions described above may lead to highly deformed transformations. To ensure that the layout of the network remains similar to the original road network, we introduce a new term into the stress function which is designed to preserve the orientations of the road segments[1]. Consider the set of all road segments $\mathcal{U}$, assuming their orientations are given by the set of angles $\{\alpha_{ij}\}_{(ij)\in\mathcal{U}}$. The double index (ij) denotes all pairs of adjacent nodes of the network. We would like to preserve the road segments orientations by minimising

$$\Psi = \sum_{(ij)\in\mathcal{U}} w_{ij}^{path}((x_i - x_j) - d_{ij}[\cos\alpha_{ij} \quad \sin\alpha_{ij}]^T)^2. \tag{12}$$

This is achieved by including the following update steps into (4):

$$\begin{aligned} x_i^{[t+1]} &\leftarrow x_i^{[t]} - w_{ij}^{path}\beta_{ang}\nabla_{alpha}(x_i), \\ \nabla_{alpha}(x_i) &= \sum_{j\in\mathcal{N}(i)} (x_i - x_j) - d_{ij}[\cos\alpha_i) \quad \sin\alpha_{ij}]^T, \end{aligned} \tag{13}$$

where summation in gradient computation $j \in \mathcal{N}(i)$ runs over the nodes j adjacent to i and β_{ang} is the weight of the penalty term Ψ in stress minimisation related to the angle of the road segments adjacent to the nodes along the path. We use the distance to the route weights w_{ij}^{path} computed as the elements of the matrix $\mathbf{W} = \mathbf{I} - \lambda_r^T\lambda_r$ with λ_r defined by Eq. (10) to allow for more flexibility around the route while preserving the original Euclidean distances towards the borders of the area of interest.

[1] Ideally one would also like to preserve the topology of the network.

3.6 Out-of-Sample Problem

The transformation method as presented above provides a mapping of every node $\mathcal{V}$ of the network from geographic space to the transformed space, $F^{\text{MDS}} : \mathcal{V} \rightarrow \mathbb{R}^M$. It does not however provide a mapping for any other locations in geographic space. This is due to the fact that in MDS, the embedded location of each sample (node) depends on it's distance from every other sample, such that all samples must be embedded at the same time. This is a limitation of MDS, known as the out-of-sample problem [29]. In order to map other features from geographical space (i.e. rivers, lakes, forests), or to transform a raster aerial photograph in the same manner as the road network, this problem must be solved (i.e. a mapping $F : \mathbb{R}^N \rightarrow \mathbb{R}^M, \mathcal{V} \subset \mathbb{R}^N$ must be provided).

The solution proposed here takes advantage of the fact that the area bounded by a set of nodes in the transformed space represents the same area bounded by the same nodes in geographic space, with potentially different shape and size. This means that a continuous non-linear function which can reproduce the mapping of nodes from geographic to transformed space should be able to map any location from geographic space, as long as the location is within the bounds of the original network. Since it is not possible to derive such a transformation function from MDS, we can instead approximate the function using an artificial neural network (ANN) of multi-layered structure which is a poweful universal approximator for empirical dependencies [30]. This is done by starting with a discrete mapping of true geographical locations $\{x_1^{geo}, \ldots, x_n^{geo}\}$ to embedded locations derived from the MDS procedure $\{x_1, \ldots, x_n\}$. The weights of the ANN are then iteratively adjusted until the ANN is able to closely approximate the discrete mapping, resulting in a continuous mapping function over the area of interest.

A raster image can then be transformed to produce a mapping of the geographic locations corresponding to each pixel in the original image to locations in the

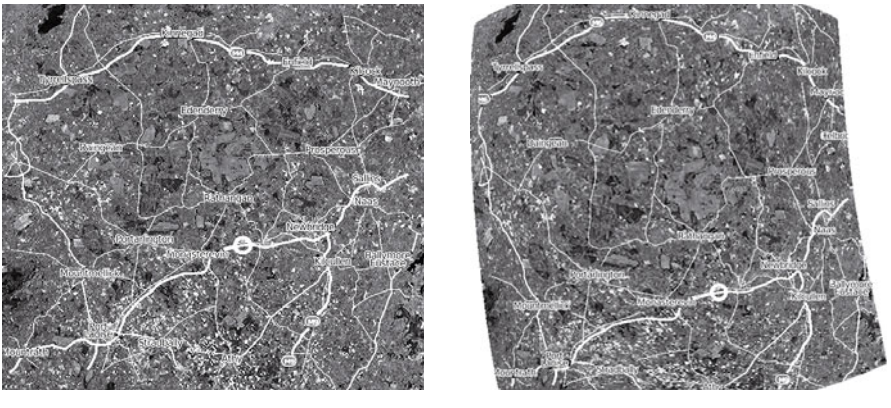


Fig. 3. Original satellite raster image (left) and its time-space representation using an ANN to approximate the modified MDS transform function(right). Note that the road network has been superimposed in both the original image and the transformed image to provide context.

transformed space using the above out-of-sample solution. Figure 3 shows this type of image transformation, and is potentially useful for navigation, as it includes features not found on the network itself, such as fields and other physical landscape features. Similarly, we are able to render vector-based map features.

4 Applications

We implemented a generalised framework for transforming geographical space based on data from the OpenStreetMap (OSM) project, within which the proposed time-distance transforms can be applied anywhere where suitable OSM data related to travel distance is available. Here, we have applied the methods covered in Sec. 3 to a series of examples based on OSM data for a region of roughly 50x50 kilometres west of Dublin, Ireland. The road network for the region includes 980 nodes (intersections), 1064 links (road-segments), and 5 different road categories. Figure 4 shows the general workflow for the time-distance map creation.

The most important stage is the introduction of time-distances for each link of the network. These distances can be provided dynamically from some other data sources as real time traffic services. For the current application however, travel times can be estimated from the network itself based on speed limits for the road categories. From this, we are able to generate a distance matrix between all points considered in the time-distance map, and feed it to the time-distance transformation, optionally followed by the out-of-sample extension. As a final step, the transformed map is rendered in a similar fashion to the original road network.

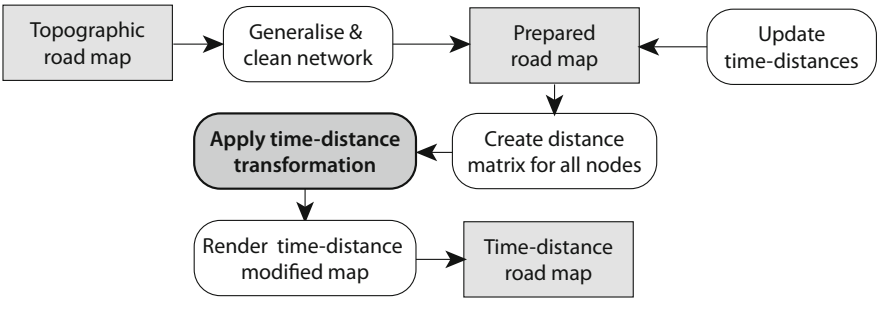


Fig. 4. Workflow for the creation of dynamic time-distance maps

4.1 User-Centric Time-Distance Map

Figure 5 (left) shows a conventional road map for the selected region, with a user-centric time-distance map (right), both rendered using the OSM style. Complete or partial topographic maps could also be represented in time-distance space, using the presented out-of-sample technique (Sec. 3.6). The time-distance representation shows the effect of the slower travel speed on the secondary roads (yellow) while motorways (depicted in blue) help reduce the time distances.

Fig. 5. User-centric traditional (left) and time-distance map created using OSM data

Note however, that the transformed map remains similar enough to the original road map to aid in interpretation. The map scale of the transformed map is based on time units around the position of the user. Places further away from the user's location are positioned somewhere between a topographic and time-distance representation to ensure an optimal interpretation and embedding into the topographic map space. However, in this case, border effects are present as no landmark nodes were defined. Normally, to embed the transformed space within a larger geographical space, one needs to define all outgoing routes at the border as landmarks, keeping the locations fixed as described in Sec. 3.3.

4.2 Route-Centric Time-Distance Map

Figure 6 shows a selected route along a traditional (left) and time-distance map (right) computed as described in Sec. 3.4. In this route-centric map, the route

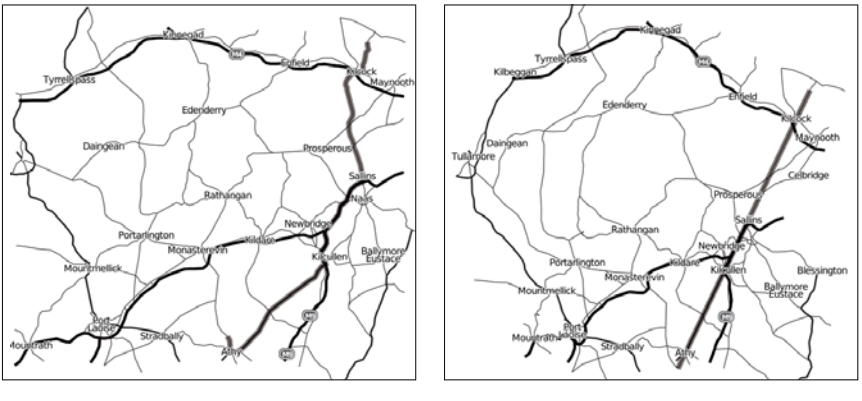


Fig. 6. Route-centric traditional (left) and time-distance map created using OSM data

becomes a straight line in the time-distance space. As such, objects surrounding the centred route are positioned according to their travel time from the route. This representation allows the mapping of additional objects along the road, such as tourist attractions, restaurants, or other facilities that may be of interest to the traveller. The user of the map has a fast, accurate view of the time needed to make a detour towards one of the surrounding features, or services. As such, the route-centric time-distance map can be a useful tool for planning trips. If desired, elements of topographic maps could also be included into such a map.

4.3 Dynamic Map Changes in the Route-Centric Representation

During a trip, the route-centric time-distance representation can potentially change constantly. If this is the case, it would be necessary to update the map dynamically, according to the current position of the user. For example, when driving down a motorway, the time-distances to the surrounding features will depend on the distance to the nearest exit. Conversely, if the user is close to the entrance to the motorway, all regions accessible via that motorway become closer. We have experimented with this idea by using a weighting scheme $\mathbf{W} = \lambda_{x_c}^T \lambda_{x_c}$ in iterative stress majorization as location x_c changes while the imaginary user progresses along the route. Figure 7 shows several steps of a route-centric time-distance representation of a hypothetical drive along various type of roads in

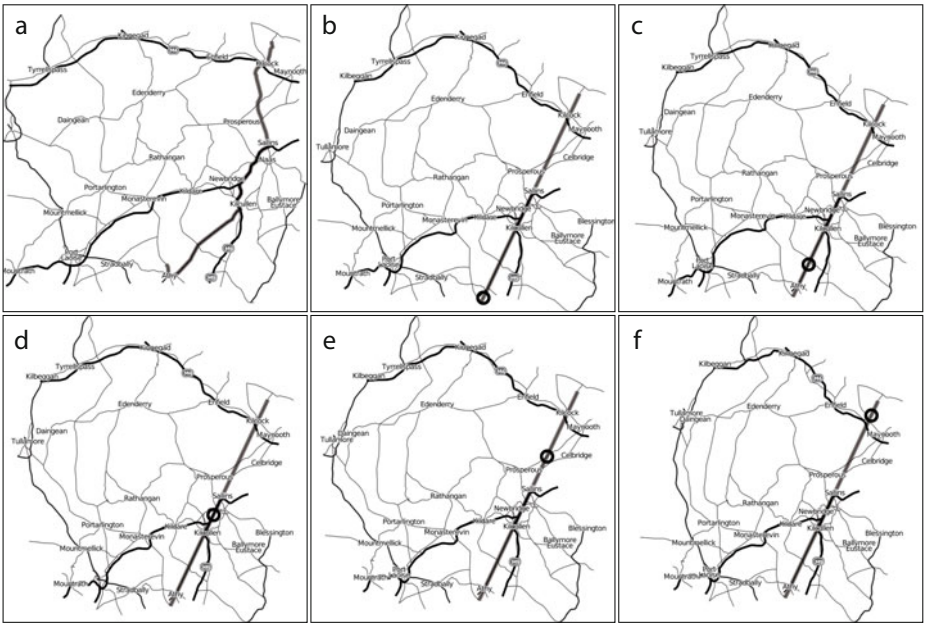


Fig. 7. Route-centric traditional (a) and time-distance maps changing during a trip (b-f)

rural Ireland. The lengths of the road segments along the straightened route exactly match the driving time spent on each corresponding segment along the path. The time-distance map shows a number of changes throughout the drive, with objects and paths further away from the route remaining relatively stable.

5 Discussion and Conclusions

This paper discusses ways of enhancing the interpretation of road maps through integration of the time dimension in terms of network travel time. We have achieved this integration through the use of new algorithms for time-distance map transformation. In general, the algorithms presented here are designed to transform the geographical space of conventional road maps into a spatial-temporal hybrid map, where locations of network nodes are adjusted to reflect the time it takes to travel between them, while still maintaining some characteristics of the conventional geographical road map such as direction and orientation. More specifically, we provide several approaches for integrating time-distance maps in a user-oriented mapping environment using real world road maps. The first approach provides a user-centric visualisation in which map features are positioned relative to their travel time from the user. As such, features that are easily accessible in terms of travel time from the user's current position are placed closer to the user. The second approach focuses on representing the temporal proximity of features relative to a specified route of path along the road network. In this case, the visualisation is route-centric, whereby the main route is shown as a straight line path between point A and point B, and all other features are positioned by their time-distance from the nearest access point along the route. The geographical extent of the applied transform and the smoothness of the transition can be easily controlled with two parameters. The values of these have to be tuned to maximise the usability of the algorithm via cognition studies aimed to adjust its performance for different applications.

There remains many interesting avenues for future work on MDS-based time-distance transformations. The algorithm presented is not designed to strictly respect the topology of the network, which presents a potential limitation. While the tools to preserve topology are available, we have not yet explored them to the level required for proper implementation. Furthermore, improvements to our proxy out-of-sample extension are required, and a hybrid approach of piece-wise linear deformations based on triangulated irregular networks and conventional rubber-sheet warping are possible avenues for exploration.

The time-distance maps presented in this paper are potentially more useful in the context of in-car map navigation than classic space-time maps, due largely to the additional constraints designed to limit the level of network deformation. However, usability tests on decision making and cognition should be conducted in the future to provide additional insights into how useful such maps are for the average user. Such tests could also provide some indication as to how much a road network can be warped and still remain usable. Additionally, it should be studied which elements could be added to a time-distance map for facilitate the interpretation.

Issues remain when working with very large and detailed road networks where the cost of the required computations can become a problem for creating dynamic maps. Algorithmically we approached the task by developing an iterative procedure to minimise an MDS stress function which we compute with a customised pair-wise distance and weights matrix. While iterative nature of the stress majorization algorithm simplifies its adaptation for the use in a temporal setting, there are still convergence issues to be explored. Further work will show if our method is of reasonable efficiency for implementation as a server-side module delivering morphed maps to a mobile device working in a dynamic environment, where traffic conditions and user location and mode of transportation are constantly changing.

Acknowledgements

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) and Stokes Programme by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

1. Karlin, O.: Time travel (2005) (Web page), http://oskarlin.com/2005/11/29/time-travel:q
2. Carden, T.: Travel time tube map, http://www.tom-carden.co.uk/p5/tube_map_travel_times/applet/
3. Böttger, J., Brandes, U., Deussen, O., Ziezold, H.: Map warping for the annotation of metro maps. IEEE Computer Graphics and Applications 28(5), 56–65 (2008)
4. Spiekermann, K., Wegener, M.: The shrinking continent: new time-space maps of Europe. Environment and Planning B: Planning and Design 21, 653–673 (1994)
5. Ahmed, N., Miller, H.: Time-space transformations of geographic space for exploring, analyzing and visualizing transportation systems. Journal of Transport Geography 15(1), 2–17 (2007)
6. Shimizu, E., Inoue, R.: A new algorithm for distance cartogram construction. International Journal of Geographical Information Science 23(11), 1453–1470 (2009)
7. Dorling, D.: Area Cartograms: Their Use and Creation. Geo Abstracts University of East Anglia, Norwich (1996)
8. Tobler, W.: Thirty five years of computer cartograms. Annals of the Association of American Geographers 94(1), 58–73 (2004)
9. Gastner, M.T., Newman, M.: Diffusion-based method for producing density equalizing maps. Proceedings of the National Academy of Sciences of the United States of America 101(20), 7499–7504 (2004)
10. Bunge, W.: Theoretical geography. PhD thesis, University of Washington (1960)
11. Tobler, W.: Map transformation of geographic space. PhD thesis, University of Washington (1961)
12. Marchand, B.: Deformation of a transportation space. Annals of the Association of American Geographers 63(4), 507–522 (1973)

13. Forer, P.: Space through time: a case study with NZ airlines. In: Cripps, E. (ed.) Space-time concepts in urban and regional models, Pion, London, pp. 22–45 (1974)
14. Kruskal, J.B., Wish, M.: Multidimensional scaling. In: Quantitative applications in the social sciences, Sage, Beverly Hills (1978)
15. Denain, J.C., Langlois, P.: Cartographie en anamorphose. Mappemonde 49(1), 16–19 (1998)
16. Yamamoto, D., Ozeki, S., Takahashi, N.: Focus+Glue+Context: an improved fisheye approach for web map services. In: 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, November 4-6 (2009)
17. Sarkar, M., Brown, M.H.: Graphical fisheye views. Commununications of the ACM 37(12), 73–84 (1994)
18. Guerra, F., Boutoura, C.: An electronic lens on digital tourist city-maps. In: Mapping the 21st century: proceedings of the 20th International Cartographic Conference, Beijing, pp. 1151–1157 (2001)
19. Torgerson, W.S.: Multidimensional scaling: I. Theory and method. Psychometrika 17, 401–419 (1952)
20. Gower, J.C.: Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika 53(3–4), 325–338 (1966)
21. Kruskal, J.B., Seery, J.B.: Designing network diagrams. In: Proceedings of the First General Conference on Social Graphics, pp. 22–50 (1980)
22. Brandes, U., Pich, C.: Eigensolver methods for progressive multidimensional scaling of large data. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 42–53. Springer, Heidelberg (2007)
23. Kruskal, J.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29(1), 1–27 (1964)
24. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. Comput. 18(5), 401–409 (1969)
25. McGee, V.E.: The multidimensional scaling of elastic distances. The British Journal of Mathematical and Statistical Psychology 19, 181–196 (1966)
26. de Leeuw, J.: Applications of convex analysis to multidimensional scaling. In: Barra, J., Brodeau, F., Romier, G., Van Cutsem, B. (eds.) Recent Developments in Statistics, pp. 133–146. North Holland Publishing Company, Amsterdam (1977)
27. Gansner, E.R., Koren, Y., North, S.C.: Graph drawing by stress majorization. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 239–250. Springer, Heidelberg (2005)
28. Borg, I., Groenen, P.J.F.: Modern multidimensional scaling: theory and applications. Springer, Heidelberg (2005)
29. Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Le Roux, N., Ouimet, M.: Out-of-sample extensions for lle, Isomap, MDS, eigenmaps, and spectral clustering. In: Advances in NIPS, pp. 177–184. MIT Press, Cambridge (2003)
30. Haykin, S.: Neural networks: a comprehensive foundation. Prentice-Hall, Englewood Cliffs (2008)

Efficient Data Collection and Event Boundary Detection in Wireless Sensor Networks Using Tiny Models

Kraig King[1,2] and Silvia Nittel[1,2]

[1] Geosensor Networks Laboratory
[2] Department of Spatial Information Science and Engineering
University of Maine
Orono Maine, United States 04469-5711
{kking,nittel}@spatial.maine.edu

Abstract. Using wireless geosensor networks (WGSN), sensor nodes often monitor a phenomenon that is both continuous in time and space. However, sensor nodes take discrete samples, and an analytical framework inside or outside the WSN is used to analyze the phenomenon. In both cases, expensive communication is used to stream a large number of data samples to other nodes and to the base station. In this work, we explore a novel alternative that utilizes predictive process knowledge of the observed phenomena to minimize upstream communication. Often, observed phenomena adhere to a process with predictable behavior over time. We present a strategy for developing and running so-called 'tiny models' on individual sensor nodes that capture the predictable behavior of the phenomenon; nodes now only communicate when unexpected events are observed. Using multiple simulations, we demonstrate that a significant percentage of messages can be reduced during data collection.

Keywords: Sensors, wireless sensor network, model, continuous phenomenon, tiny models, process modeling, prediction, autonomous.

1 Introduction

As the field of geosensor network research matures, the number of sensor networks deployed to collect data for geospatial phenomena is increasing. This trend is spurred by significant advances in wireless communication, the miniaturization of computing and storage hardware, as well as advances in sensor materials and technology [1]. Independent networks of sensors nodes are frequently deployed to observe and monitor the characteristics of an event. These characteristics are often comprised of dissimilar measurands of the target phenomena, which can span both 3D space and time. Consider for example, monitoring the intensity of light over a finite region, not necessarily a geographic region but for example an indoor space, which is illuminated by a controlled light source. The phenomenon of the light distribution follows an expected physical process, which can be captured in a formal model. This work investigates strategies to minimize data collection in the sensor network, as sensor nodes do not need to exchange information with other nodes if the predicted process proceeds as expected. Thus, instead of communicating the 'obvious', the sensor network only

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 100–114, 2010.

initiates wider-spread activity in situations where there are measured deviations from the known model (for example, if an additional light source is added).

A key challenge in developing a model-based decentralized monitoring strategy, is satisfying the storage and computing requirements that most large models will necessitate. These models should not exceed the 416Mhz and 32MB of storage provided by some of the most advanced sensor hardware offered today [2]. The goal of this research is to map a predictive, often large and complex model to a set of "tiny models", which can be run on Micaz sensor nodes with limited memory and processing resources [3]. These tiny models will provide each node with sufficient knowledge to evaluate the expected process based on time and their spatial location with regard to the phenomena being observed.

Our objective is to minimize overall data communication and reduce it to handling unexpected values. In such an event, sensor nodes will reason about the cause of the deviation; possible causes may include noisy sensor readings, an observed event, or incomplete (perhaps even inaccurate) model information.

Streams of sensor readings represent 'snapshots' of objects and processes that continually change over time. Models are generated from previous knowledge of how sensed objects and/or processes evolve over some temporal period, and are used to assist nodes in understanding and monitoring evolutionary changes in the network. An event is considered a record of a process change of interest, that is, a measured deviation from the known process model at some fixed time [4]. For example, a model may predict that light intensity at a measured location should be similar (e.g. within +/- 0.1 W/m^2) to spatially and temporally adjacent sensor readings. In this scenario, the deviation is the reported event, the location the object of interest, and the sudden change in illumination an abnormal process state. Relative to the known process model, the sensed event, object of interest, and abnormal process state become the phenomena of interest, which initiates further node communication to reason about the abnormal sensor data.

The next section introduces the research problem. The remainder of this paper is structured as follows: a brief motivational example of how a prototypical model can assist in minimizing radio communication in a sensor network which is followed by a discussion of the hardware constraints and operational requirements. This will become a primer to a detailed study of methods for developing smaller models from detailed parent models that are complete and thus much larger. A formal framework and rationale is then introduced, that utilizes these 'tiny' models. We show that our strategy for decentralized areal estimation using tiny models will yield efficient, semi-autonomous sensor networks by leveraging and evolving models of an understood process on resource constrained hardware. Finally, conclusions and plans for future work are discussed in the final section of this paper.

2 Predictive Model-Based Data Collection in Sensor Networks

Scientists and engineers are frequently interested in monitoring an understood phenomenon in order to verify process stability, to identify the occurrence of abnormal events, and foremost be alerted to them. For example, engineers may want to identify abnormal machine vibrations to assist in predicting mechanical failure. Today, we can use wireless sensor nodes to autonomously monitor these phenomena at novel spatial

and temporal scales. However, even the most state-of-the-art hardware still has limitations such as power consumption, storage capacity, and processing capability. In particular, this applies to the application of sensor networks for the observation of well-known phenomena whose process can be captured in predictive models. Traditionally, sensor nodes are used for the raw collection of data, which is then transmitted to a central base station for verification and comparison to a forecast phenomenon. Alternatively, communication costs can be significantly reduced if the sensor nodes could autonomously and intelligently make predictions, detect abnormal values locally, and only communicate alerts in exceptional cases.

To accomplish this, we propose breaking up traditional large, complex predictive models into 'tiny models' and loading these compact models of the target phenomena onto individual sensor nodes. By executing tiny models locally on sensor nodes, it is likely that a significant decrease in communication cost can be achieved due to a reduced need for transmitting raw sensor data readings to other nodes and throughout the network. Communication activity is limited to 'unusual' events by avoiding 'obvious' data collection and instead only initiating wider-spread sensor activity where measured deviations from the known model exist. Sensor readings are autonomously and locally compared to derivations from a tiny model to isolate unpredicted observations, which are further discriminated as sensor noise, environmental noise, or an actual event of interest. For example, a sensor could measure a noisy value or indeed an (unexpected) event. Local collaboration is reduced to nodes only interacting with neighboring nodes to analyze the cause for the deviation. In the case that an event happens, likely the neighboring nodes will sense similar values, and the boundary of the 'event' can be computed. The remainder of this paper explores various techniques for the creation and dissemination of a 'tiny model', methods for efficient boundary detection using this model, and techniques for in-network data suppression.

2.1 Example

Let's assume that the process being observed, for example the spatial distribution of light in an observed region over time, can be represented by a mathematical model of the variance in illumination intensity at any temporal snapshot T. All nodes are programmed with a minimalistic variation of this model, and once deployed, nodes persistently sample the spatial region of interest at a discrete interval that is appropriate to accurately observe estimated changes in light intensity (for example, at a frequency that is 1/100th the expected rate of change.) This observational process can be decomposed into three transition states, each of which progressively demands more node-node communication. The first state, field monitoring, is characterized by the node performing predictive sampling/validation of the environment relative to the model, requiring the least communication with neighbors. The second state, event detection, draws upon knowledge from neighboring nodes to identify events of interest, employing a balance of internal model validation and cross-comparison with neighbors. The third state, event contour processing, requires the highest level of node collaboration to accurately monitor the event boundary.

Our objective for using tiny models is that they will permit nodes to collect data samples locally and then cross-validate this data with the model prediction. Ideally, the model will facilitate independent operation of the nodes and no radio transmission

within the network would be required. However, due to spatial knowledge requirements nodes may still need to periodically communicate with each other in order to synchronize the model with reality, validate on-board sensor readings, and be responsive to neighboring requests for validation. That is, intermittent readings from neighboring nodes are necessary to refresh model predictions and also to determine if a deviation from these calculations is due to noise or some unpredicted event. In the latter case, a sensed deviation from ambient light that exceeds a predefined limit for some temporal period will trigger further processing within the network, commanding all neighboring nodes to cross-compare their sensed values of the field and collaborate to determine potential noise in the measurement or the extent of aberrations from the prediction.

To set the groundwork for a prototypical example, let's assume that twelve nodes are arranged in a fixed 3 x 4 spatial grid to monitor light intensity along the surface of a room. To simplify the illumination function, it is assumed that there is a stationary point source of light, centered in an empty room, at a fixed distance above the floor. These design constraints permit the use of a simple illumination function, which takes as input the power and location of the light source, its distance above the floor, and the measurement location. For a single light source this function is formalized with the following equation:

$$I = W / (4*pi * ((M_x - S_x)^2 + (M_y - S_y)^2 + D^2)) . \quad (1)$$

Where: I = illumination, Watts/m^2
W = bulb power, Watts
$M_{(x,y)}$ = Cartesian coordinates of measurement location, m
$S_{(x,y)}$ = Cartesian coordinates of bulb location, m
D = distance of the bulb above the floor, m

The formula above models illumination along the base of a room from a single source of light. For example, we consider a 60 watt bulb 3 meters above the floor and positioned at room center (i.e. S_x =0, S_y =0). The sensor nodes are equipped with a localized program that predicts the illumination based on the node's spatial location (i.e. M_x =0, M_y=0). We call this a 'tiny model' since it can be stored and computed on a resource constrained sensor node. Communication activity is minimized by locally sampling the light intensity at a discrete point and then comparing it with the internally executed model. This sampling and comparison occurs at an interval sufficient enough to accurately capture an event, for example, a sudden drop in illumination within the monitored environment. If this is the case, it initiates an in-network decentralized algorithm, which draws upon knowledge from neighboring nodes to determine the cause of the deviation, potentially isolating an event of interest. For example, if neighboring nodes do not sense a similar deviation, it is likely that the deviant reading is attributable to sensor or environmental noise. However, if a number of adjacent nodes detect similar departures from their own tiny models, more frequent and targeted sensing within this region of interest begins and data is communicated upstream to a central node for further analysis and estimation of the event's areal extent.

2.2 Requirements

The objective of this research is to minimize communication within the sensor network by verifying process observation locally, and by leveraging knowledge that is traditionally captured in large complex models of an understood process within more compact 'tiny models'. These models need to be sufficiently constrained such that they can be executed on individual nodes within a sensor network. Characteristics that are considered include the hardware constraints of RAM, Flash, CPU, and power, as well as how system accuracy and precision are affected by design decisions such as the spatio-temporal sampling frequency of sensor data. This novel approach, known as 'TinyModeling', is expected to provide an energy reduction approaching 85% over raw data collection. This energy reduction is derived from a measured reduction in the number of messages transmitted during process observation and event detection. Message transmission is the primary consumer of a node's hardware resources, thus, a reduction in sensor network communication offers significant opportunities for resource conservation.

MEMORY/CPU FOOTPRINT: A major challenge in designing tiny models is how to capture knowledge with models that are based on point samples (or close neighborhoods) and that must run in a severely hardware constrained environment. Alternatively, traditional 'large' models run on powerful CPUs with large reference data sets, comparatively unlimited persistent storage, and substantial RAM. To better understand the operational boundaries for tiny models, one must consider the overhead required by a node operating system such as TinyOS. For example, the TinyOS kernel only occupies approximately 400bytes of storage while the required nesC runtime primitives and radio interface use another 3.1Kb [5],[6]. This lightweight operating system retains a significant portion of the sparse resources available for storage of the tiny model and associated algorithms. For example, a MicaZ mote running TinyOS provides approximately 3.7Kb of RAM, 6.1Mhz of available CPU duty, and 124.6Kb flash memory for application programming. As can be seen from Figure 1, on most current node platforms at least 80% of all available resources can be used for tiny models. One can also expect a significant conservation of power by implementing tiny models rather than raw data collection and communication schemes. For instance, at peak load a MicaZ node consumes approximately 19.7mA of current

DCP RESOURCE AVAILABILITY

	RAM		CPU		PROGRAM FLASH	
DCP	AVAILABLE	REMAINING	AVAILABLE	REMAINING	AVAILABLE	REMAINING
MICAz	4Kb	3.7Kb	8Mhz	6.1Mhz	128Kb	124.6Kb
IRIS	8Kb	7.7Kb	8Mhz	6.1Mhz	128Kb	124.6Kb
Imote2	256Kb	255.7Kb	416Mhz	414.1Mhz	32Mb	31.9Mb
TelosB	10Kb	9.7Kb	8Mhz	6.1Mhz	48Kb	44.6Kb
TmoteSky	10Kb	9.7Kb	8Mhz	6.1Mhz	48Kb	44.6Kb

Fig. 1. Comparison of data collection platform (DCP) resource availability when running TinyOS

while transmitting and receiving messages. However, without external communication the node only utilizes 8mA of current, conserving up to 60% of the available power during deployment.

ROBUSTNESS/ACCURACY: The system of nodes and 'tiny models' run within a specific context: we expect to observe a likely and anticipated process most of the time, and only need to occasionally measure this process to verify that the phenomenon is proceeding according to the model. However, the nodes must accurately and quickly, detect and analyze the unexpected, classifying it as either noise or an event, and only performing sporadic sampling and verification (to avoid false positives).

The objective of developing tiny models is not to capture perfect knowledge of the observed process over some temporal period. We pose the accuracy of traditional complex models at the targeted precision; however, we assume that the compact models will be less precise. Tiny model implementation in its most compact form, must take an aggressive stance on balancing internal data collection and storage with external communication among neighboring nodes. A sufficiently significant data set, one that provides adequate temporal as well as spatial density, is required to meet the desired accuracy and precision requirements for trending and analysis. Model-based areal event detection targets the 'exceptional' case and not the 'norm', therefore, communication with neighboring nodes should only be instantiated for further clarification and examination of unexpected data. Depending on the architecture of the tiny model, periodic local collaboration between networked nodes must occur to verify sensor readings and perform model validation; however, this does not provide optimum performance. Strategies such as alternating wake and sleep cycles among nodes, as well as leveraging staggered local clocks instead of global synchronization can assist in minimizing power depletion due to communication.

3 Designing Tiny Models

The main goal in designing a tiny model is to sustain as much intra-node data processing as possible, in order to minimize network collaboration (and expensive communication) about obvious events. However, periodic node-node communication is required for model dissemination and is critical for process observation once an event has been detected. To facilitate node-node communication, an adequate network topology must exist. Although mobile WSNs offer a number of interesting challenges, currently we focus on a static network to develop and test the TinyModeling approach. We assume a sensor network is arranged as a collection of N_i sensor nodes, located in a uniform grid pattern defined by C_x columns and R_y rows (Figure 2). In order to avoid orphaned nodes, it is assumed that each node is positioned so that it can collaborate with at least one neighbor. That is, the distance between spatially adjacent nodes must not exceed the reliable transmission range of the hardware being used. Such a topology simplifies the algorithms used for location aware models by minimizing distance variation in the spatial distribution of sensors, allowing the communication paths between nodes to be predictable.

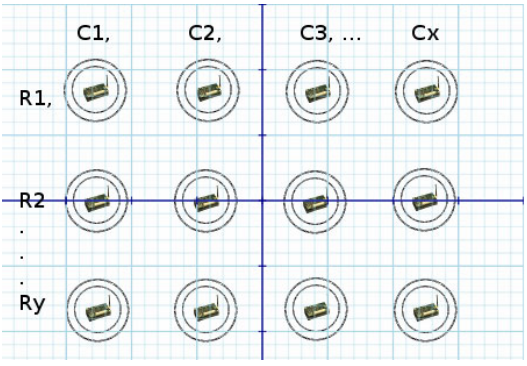


Fig. 2. A prototypical sensor network composed of C_x columns and R_y rows of Mica motes

Using this topological structure as our test bed for deployment of up to 289 sensor nodes, we introduce a first mechanism for disseminating prior knowledge of a phenomenon (e.g. the illumination model discussed earlier) to individual sensor nodes within the network. Our interest is in 'mapping' large complex models into tiny models that can be run on individual sensor nodes. In the following research, we test our approach using a well-understood physical process (light distribution from one or more light sources) that we can capture in an equation and simulate as a process. The objective is to compute the overall model, initialize the sensor network with these tiny models, and then test the model-based process (observation). Overall, the experiment consists of several steps: (1) setting up a communication topology to initialize the nodes with the tiny models, (2) initializing individual nodes with their tailored tiny models, (3) process observation and (4) in-network localized event handling.

Starting with step 1, a single node is designated as the base station (i.e. root) and it is preloaded with the large complex model of the phenomenon being observed. It is assumed that the base station node is a line-powered device, connected directly to a PC via a physical connection such as serial or USB. At initialization of the sensor network the root coordinates communication between all nodes, helping to assert the routing protocol that will be used for future upstream and downstream communication among nodes. A discussion of these communication strategies is outside the scope of this paper, however, we encourage readers to review established techniques such as tree, star, and clustered topologies. For this research, the multi-hop tree collection schema provided by TinyOS has been utilized to flow messages to and from the root.

Upon successful self-organization of the sensor network, with regard to the initialization routing tree, the root node queries the spatial location of all other nodes. These coordinates are fed into the larger, complete model, with the resulting output being the coefficient(s) describing the phenomenon at the prescribed node location. This information becomes the basis for a 'tiny model', which will be transmitted back to the node and stored for future in-network sensor validation. For example, let's assume a node at location (2, 4) transmits its position to the root of the network tree. The root then calculates the intensity coefficient of the light at the node's reported location. Assuming a ninety watt bulb located at the center of the room (0,0) and positioned three feet above the floor, the calculation per equation (1) is:

$$I = 90 / (4*pi *((2 - 0)^2 + (4 - 0)^2 + 3^2)) .$$
$$I = 0.247 .$$

The resulting coefficient is transmitted back to the node and becomes the basis for the most simplistic type of tiny model, a single coefficient, which describes the anticipated sensor reading at the current measurement location. This process repeats iteratively until each sensor node has received its own miniaturized version of the larger model from the root node (albeit in this case only a coefficient). Thus, this framework permits model prediction at all node locations.

Upon receipt of the tiny model definition, phase 3 starts, and each sensor node is released to begin internal data acquisition and autonomous comparison to the predicted nominal value. Should the sensor value remain within a predetermined upper and lower tolerance band (i.e. upper spec limit, USL; lower spec limit, LSL) about this nominal, the node's radio remains off and it is assumed by the root that the process (e.g. illumination) remains compliant (Figure 3a). Alternatively, the sensed value may drift outside of the permissible tolerance thresholds, indicating the presence of data uncertainty or the occurrence of some event (Figure 3b). Before either explanation of variation is considered plausible, additional analysis must first occur to deduce the cause.

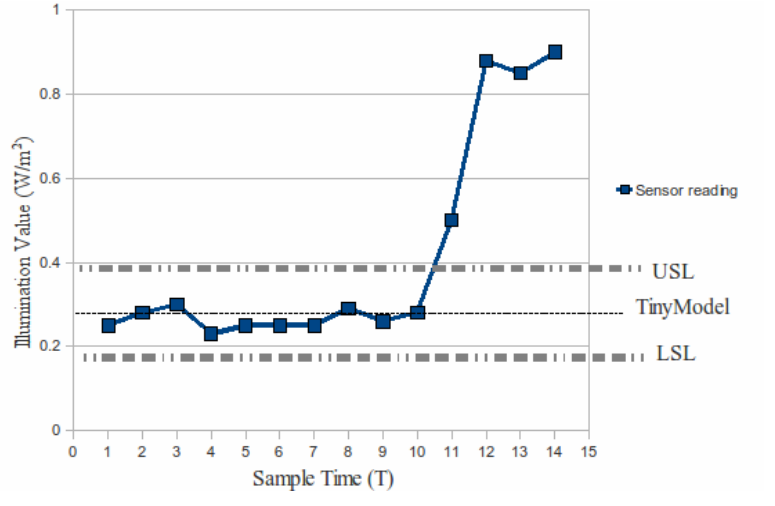


Fig. 3. The graph above demonstrates sensor values and their various states of adherence to predictions made by a tiny model of the observed phenomenon, represented here, by the dotted line at the ordinate 0.3. Thus the process can be described as (a) compliant, readings 1-10 and (b) uncertain, readings 11-14.

4 Event Detection Based on Tiny Models

During the comparison of sensed data to the tiny models' prediction, nodes must be empowered to classify the detected result as either: (1) a non-event, i.e. the process

behaves as expected, (2) uncertainty due to environmental noise, (3) faulty readings due to sensor failure, or (4) an event that requires further action to isolate its boundary. We define an event explicitly to be drift from the predictive model, which is characterized by a discrete spatiotemporal dimension. This value is defined by the system user, and quantifies the threshold between noise and event detection. For example, if the sensed illumination suddenly exceeds a permissible threshold (e.g. the upper specification limit -USL) beyond the model prediction, the node must have the necessary logic and resources to reason about the cause of the drift. Possible events may be an object passing through the network, which induces a shadow, or the introduction of a second light source, which increases the light intensity at specific node locations.

Once a significant departure from the tiny model has been identified, the node attempts to mitigate the cause autonomously. This is achieved by comparing the non-compliant sensor reading to an internal cache of historical sensor readings: $\{R^{-1}, R^{-2}, R^{-3}, \ldots R^{-n}\}$. The depth of the cache, **n**, must be sufficient enough interpolate trends which may be indicative of an event, but not so large that it consumes excessive hardware resources. This system parameter is highly dependent upon the temporal sampling schema, as well as the expected rate of change of the phenomena being measured. For instance, the illumination example may be understood to react as an approximate binary process, with a rapid and near constant drift from the expected value (e.g. a new light source is suddenly added or the original one goes dark). In such a scenario, a cache size of ten historical values may provide sufficient resolution to flag a recurring departure from nominal that requires additional investigation.

For example, if the current sensor reading $\{R^{0}\}$is non-conforming but the previous ten readings $\{R^{-1}\ldots R^{-10}\}$ match the prediction, the node assumes the uncertainty is attributable to environmental noise and no further action is taken. However, should a statistically significant number of these historical readings also exhibit a similar departure from nominal, further processing will be initiated to identify if the cause is a faulty sensor or the existence of an event. Because the node has no additional internal knowledge available for reasoning, it must initiate radio communication within the sensor network to query if neighboring nodes have experienced similar departures from their own model predictions. If no neighboring nodes detect such departures from the model, the node assumes that the abnormal reading is due to a faulty sensor. It continues to analyze future data acquisitions, but waits a defined number of measurement cycles before it again queries neighboring nodes for the existence of confirmed model departures.

Alternatively, neighboring nodes may return information that they *have experienced* a similar model departure. When such a confirmation occurs, nodes self-organize to detect the extent of the occurring event. Based on previous work [7], [8] an energy efficient algorithm is used to identify only the boundary of the areal event and track its changes over time. Each node communicates with its direct neighbors to identify if it is located on the boundary of the event or "inside" of the event. In the case of being a node located 'inside' of an event's region, all neighboring nodes show similar derivations from their models. In this case, the node stops communication again, and resumes regular local sampling. If a node, however, identifies that several of its neighbors experience no model derivation, but others do, it can identify itself as a boundary node and identify potential other neighboring boundary nodes.

Once all sensor nodes have identified themselves as either boundary nodes or 'inside' event nodes, the boundary nodes continue communication in regular intervals to observe the event boundary and its changes. Successive monitoring of the event boundary is only performed by these boundary nodes, and the one closest to the root is elected to transmit an aggregated list of the nodes which compose the event boundary. This decentralized algorithm permits efficient boundary estimation, minimizing the number of messages required to monitor the event. To improve the resolution of the sensor network, the boundary nodes may also increase their sensor sampling frequency in order to monitor the event with a higher precision. Should any node begin acquiring a consecutive number of sensor readings that are within normal operating parameters, it will stand down, and cease to transmit data until a future event is sensed (using the procedure prescribed above). This protocol permits nodes to minimize unnecessary radio communication, by only transmitting data when a confirmed event has been detected.

5 Performance Evaluation

The tiny model framework consists of several algorithmic parts: (1) initializing the nodes with their models, (2) continuous observation, (3) noise, event detection and identification. Since the initialization is run only once, or rarely (e.g. to recalibrate), we assess the communication cost for this part separately. The major part of the performance testing is done with regard to steps two and three. We expect that phenomenon observation without event detection is the most interesting part of the performance analysis since we foresee the highest energy savings here. Event detection and handling is similar to other approaches in this research area (e.g. boundary detection algorithms); however, it is performed based on tiny model information. In our experimental setup we test the communication cost for all three parts, and compare the observation with raw sensor data collection and tree based routing as a baseline.

For simulation and testing purposes, three 4m by 4m grid topologies of different sensor node 'resolution' were constructed (see Figure 4). The first is composed of 25 nodes with a 2m spacing. The second contains 81 nodes with a 1m spacing and the

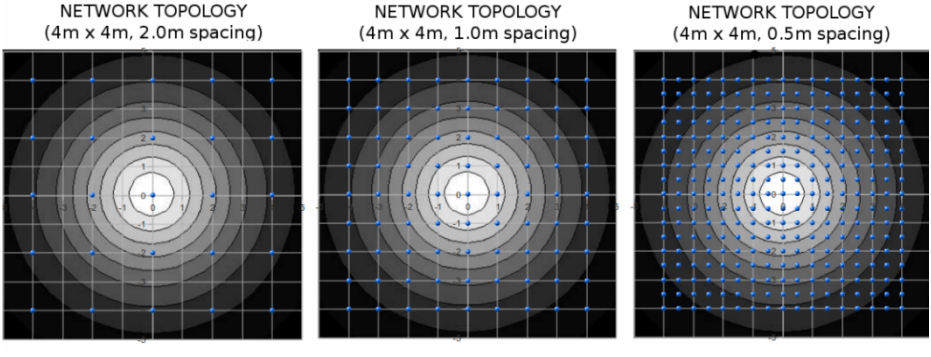


Fig. 4. Prototypical sensor topologies used to measure the illumination gradient for a light source located at (0,0). (a) 2m spacing, (b) 1m spacing, (c) 0.5m spacing.

most dense configuration contains 289 nodes with a 0.5m spacing. Each of these topologies has a relaxed signal to noise schema to facilitate predictable communication (i.e no delayed or lost messages) among all nodes in the network. This eases retransmission of messages due to signal variance and path loss, permitting a more accurateperformance evaluation of event detection utilizing the tiny model framework. The varying choices of sensor network density are necessary to assess the detection accuracy.

To organize the nodes, a multi-hop, tree topology-based network configuration is employed. The simulation is executed for twenty-four hours with sensor data sampling occurring at five minute intervals, testing the three different network densities. In the case, of raw sensor data collection, the sensor data is sampled and sent upstream to the base station without further aggregation. Analyzing tiny models, we test several sensing strategies to quantify the number of messages TinyModels require, and compare this with the number of messages in the raw sensor data collection case.

In the *first test*, we determine the simulation baseline for the TinyModels approach; here, we assume the tiny models are disseminated to the nodes once, and no further events, noise or even synchronization takes place. In the *second test*, we add a synchronization beacon to the TinyModel protocol, requiring all nodes to transmit a confirmation of functionality to the root every hour or two hours. In the *third test*, we additionally factor in that a percentage of the nodes experience sensor noise locally, which is characterized by internal sensor data exceeding model predictions but having no correlation with readings from neighboring nodes. This noise occurs for a finite period of time, at a frequency of once per hour. More specifically, scenario 3a tests 20% of the nodes experiencing noise while test 3b five assumes 40% of the nodes experience noise. In the *fourth test*, we select a set-up that uses hourly synchronizations of nodes with regard to the tiny models and a 20% noise ratio. It should be noted that unclassified noise, which has both a spatial and temporal extent may be misconstrued as an event if clusters of nodes experience similar noisy data. Such false positives for event detection require the system user to later classify these occurrences as either events or noise. Additionally, we assume event detection over a discrete spatial area within the sensor network covering about a sixteenth of the overall observation area. We assume that three events are detected (at hours 6, 12 and 18). It takes one cycle to detect an event, and can take several cycles to observe the event depending on its duration. In the simulation, the 6^{th} hour event is detected during a single five minute collection cycle; event detection spans four collection cycles during the twelfth hour, and two collection cycles during the eighteenth hour. The results of each simulation are shown in Figures 5 and 6.

These test scenarios demonstrate that the TinyModel approach significantly reduces sensor network energy consumption during routine monitoring of a well understood phenomenon. Most notably, the number of messages for the baseline scenario of raw data collection in a 289 node topology was reduced from 521,280 to 46,583 messages using TinyModel-based event detection. This is due to the dissemination of intrinsic process knowledge that empowers network nodes to autonomously reason about the sensor data they acquire. The creation of tiny models, coupled with targeted distribution of this knowledge, enable in-network data evaluation that minimizes radio communication with the root node when the process is operating as expected. The results for each simulation and the associated reduction in messages are shown in Figure 7.

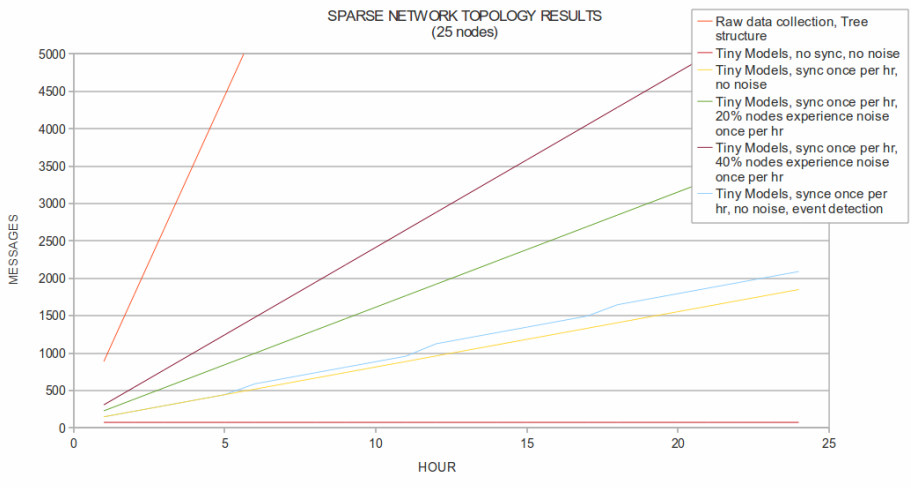


Fig. 5. Raw data collection vs. TinyModeling test results for a sparse topology (e.g. 25 nodes).

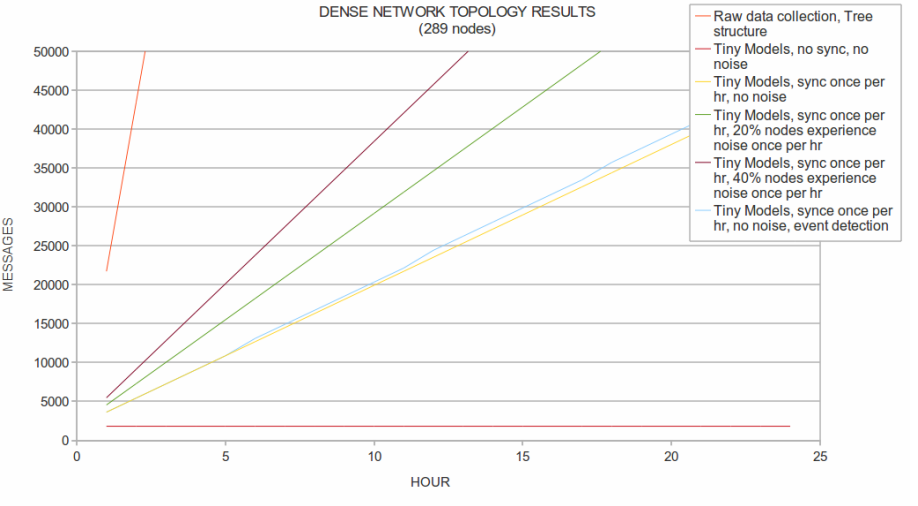


Fig. 6. Raw data collection vs. TinyModeling test results for a dense topology (e.g. 289 nodes)

NUMBER OF MESSAGES TRANSMITTED (24hr simulation)						
	25 NODES		81 NODES		289 NODES	
	messages	reduction	messages	reduction	messages	reduction
BASELINE, raw data collection	*21312*	-	*105408*	-	*521280*	-
TinyModel, no sync	74	-99.65%	366	-99.65%	1810	-99.65%
TinyModel, with sync	1850	-91.32%	9150	-91.32%	45250	-91.32%
TinyModel, with sync, 20% noise	3770	-82.31%	15371	-85.42%	67445	-87.06%
TinyModel, with sync, 40% noise	5690	-73.30%	21592	-79.52%	89640	-82.80%
TinyModel, with sync, events	2091	-90.19%	9659	-90.84%	46583	-91.06%
	AVG:	-87.35%	AVG:	-89.35%	AVG:	-90.38%

Fig. 7. Simulation Results: an analysis of messages transmitted per 24 hour simulation for each of six testing scenarios and three network topologies.

Based upon these results, TinyModel-based event detection benefits mostly those processes which are assumed to be relatively stable and that frequently operate within the prescribed specification limits of the model. Given such a process, tiny models significantly reduce overall network communication, permitting node radios to remain off, conserving both power and hardware resources.

To enable more extensive testing of the hypothesis that TinyModels for a known process significantly decreases communication costs during event detection, we have developed a prototypical sensor network using TinyOS and the TOSSIM simulator. This system is sufficiently modular that additional knowledge models can be inserted into the simulation, as well as various communication protocols, and sources of noise. Additionally, the temporal frequency and spatial extent of events can be altered to test the robustness of the TinyModel framework. It is anticipated that continued testing and development will facilitate additional enhancements, which will further improve the performance of TinyModel event detection.

6 Related Work

Due to a node's limited power capacity and the high cost associated with wireless data transmission, improving communication efficiency between networked sensor nodes has been an active area in geosensor research [9]. Many strategies have been introduced to make geosensor networks more efficient by decreasing node-to-node communication and the associated transmission costs. For example, data-centric routing, draws upon an in-network analysis of individual sensor readings to permit nodes to evaluate whether or not sensed data should be sent or received [10]. For instance, nodes may choose to only power on their radios if a message of interest (e.g. that of a detected event) should be transmitted to neighboring nodes [11].

Models of an understood process are also an integral part of ongoing research that aims to advance structural health modeling using sensor networks [12]. In this application, groups of sensors are distributed throughout an engineered structure (e.g. building or bridge) to monitor vibrations that may compromise the structure's safety or useful life. Groups of sensor nodes are strategically paired to process a single structural analysis algorithm in a coordinated manner. Nodes locally process sensor readings and the collective leader transmits an aggregated set of data, or recommended model adjustment parameters, back to a reference node. Our Tiny Model strategy further decreases the volume of messages exchanged between nodes by empowering each node to autonomously analyze sensor data, and only initiate radio communication if values deviate from known model predictions.

Another field of geosensor research, tracking the patterns of moving objects, utilizes models to efficiently track the current motion of an object as well as predict future movement of the objects. One such example leverages materialized and non-materialized trajectories to improve sensing efficiency and overcome location imprecision due to uncertain data.. In addition to using the road network as a source of knowledge about the phenomena being measured, more robust moving object modeling techniques also consider velocity changes of the moving objects being sensed [12].

Within the sensor network and database communities, models have been proposed to assist with user-based queries for data acquisition in sensor networks [13]. Sensor

readings are supplemented with knowledge from predictive approximation models to augment the need to collect data from all sensors within the network. These strategies typically use statistical modeling techniques to account for issues in spatial sampling by extrapolating missing or faulty sensor data. In this paper, instead of model predictions being generated by a central coordinator and issued to downstream child nodes, we propose empowering all nodes with a miniaturized version of the process model.

7 Conclusions and Future Research

This research is different from previous work in the area of efficient sensor networking in that it utilizes localized tiny models to perform energy efficient data collection and boundary analysis of events, by leveraging characteristics of an understood phenomenon to achieve process monitoring and unexpected event detection similar to that realized with a fully detailed model. Comparing locally sensed data to the Tiny Model prediction, nodes must be empowered to classify the detected result as either: (1) a non-event, i.e. the process behaves as expected, (2) uncertainty due to environmental noise, (3) faulty readings due to sensor failure, or (4) an event that requires further action to isolate its boundary.

When a process behaves as expected, nodes only perform predictive sampling/validation of the environment relative to the model, requiring the least communication with neighbors. If deviation from the model occurs, nodes employ a balance of internal validation and cross-comparison with neighbors to isolate sensor noise and detect event boundaries. Using a simulation approach, this work has demonstrated that 'TinyModeling' is able to provide an energy reduction exceeding 85% over raw data collection (measured in transmitted messages). Additional testing will quantify the accuracy and precision of boundary detection for events, by varying the spatial density of the sensor topology and the temporal sampling schema. Future research will consider methods for achieving immunity to sensor failure, reasoning about abnormal event detection, model-evolution using back propagation, and the implications of mobile nodes.

References

1. Nittel, S.: A survey of geosensor networks: advances in dynamic environmental monitoring. Accepted for publication: Sensors Journal (2009)
2. CrossbowTech: Imote2, `http://www.xbow.com/Products/productdetails.aspx?sid=253` (Visited 10. 04. 2009)
3. CrossbowTech: Micaz, `http://www.xbow.com/Products/productdetails.aspx?sid=164` (Visited 10.04.2009)
4. Galton, A., Worboys, M.: Processes and events in dynamic geo-networks. In: Rodriguez, M., Cruz, I., Levashkin, S., Egenhofer, M.J. (eds.) GeoS 2005. LNCS, vol. 3799, pp. 45–59. Springer, Heidelberg (2005)
5. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. ACM SIGPLAN Notices 35(11), 93–104 (2005)

6. Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., Culler, D.: TinyOS An operating system for sensor networks: Ambient Intelligence, vol. 2, pp. 115–148. Springer, Heidelberg (2005)
7. Duckham, M., Nittel, S., Worboys, M.: Monitoring dynamic spatial fields using responsive geosensor networks. In: ACM-GIS 2005, Bremen, Germany (2005)
8. Jin, G., Nittel, S.: Supporting spatio-temporal queries in wireless sensor networks by tracking deformable 2D objects. In: ACM-GIS 2008, Los Angles, CA (2008)
9. Stefanidis, A., Nittel, S.: GeoSensor Networks, p. 296. CRC Press, Boca Raton (2005)
10. Huang, H., Hartman, J., Hurst, T.: Data-centric routing in sensor networks using biased walk. In: 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks(SECON 2006), vol. 1, pp. 1–9 (2006)
11. Ditzel, M., Langendoen, M.: D3 Data-centric data dissemination in wireless sensor networks. In: European Conference on Wireless Technology, Paris, France, October 2005 (2005)
12. Nagayama, T., Spencer, B., Agha, G., Mechitov, K.: Model-based data aggregation for structural monitoring employing smart sensors. In: Proceedings of the Third International Conference on Networked Sensing Systems (INSS 2006), May 31- June 2, pp. 203–210 (2006)
13. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., Hong, W.: Model-driven data acquisition in sensor networks. In: Proceedings of the 30th International Conference on Very Large Databases (VLDB), Toronto, Canada, vol. 30, pp. 588–599 (2004)

Combining Synchronous and Asynchronous Collaboration within 3D City Models

Jan Klimke and Jürgen Döllner

Hasso-Plattner-Institute, University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, Potsdam, Germany
{jan.klimke,juergen.doellner}@hpi.uni-potsdam.de

Abstract. This paper presents an approach for combining spatially distributed synchronous and asynchronous collaboration within 3D city models. Software applications use these models as additional communication medium to facilitate communication of georeferenced and geospatial information. Collaboration tools should support both the communication with other collaborators and their awareness of the current collaboration context. To support collaborative knowledge construction and gathering, we have designed a collaboration system to facilitate (a) creation of annotations that have 3D references to the virtual 3D city model and (b) collection information about the context in which these annotations are created. Our approach supports synchronous collaboration in connection with the creation of non volatile, precisely georeferenced units of information allow for a comprehensible form of cooperation in spatially distributed settings. Storage and retrieval of this information is provided through a Web Feature Service, which eases integration of collaboration data into existing applications. We further introduce a visualization technique that integrates annotations as complex structured data into the 3D visualization. This avoids media breaks and disruptions in working processes and creates a spatial coherence between annotation and annotated feature or geometry.

Keywords: Collaboration, Geospatial Annotation, 3D Geovirtual Environment, 3D Information Visualization.

1 Introduction

Virtual 3D city models represent complex urban geographical and geospatial data. A number of systems provide functionality for presentation, exploration, analysis, and management of these models. Application domains such as urban planning, landscape architecture, city marketing, tourism information, and disaster management, typically involve a large number of stakeholders, specific requirements, and appropreate work flows. Users from different areas of expertise need to work collaboratively to master the complexity evolving from these issues. The need for communication between these experts is rising due to the increasing level of specialization and internationalization of parties, each working on specific aspects of a project. Since virtual 3D city models as *3D geovirtual*

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 115–129, 2010.

environments (3D GeoVEs) serve as integration spaces for multiple, heterogeneous, and distributed sources of geodata [1], they are designated to simplify communication of space-related information. In contrast to other environments for collaboration, e.g., maps or text-based systems, 3D GeoVEs are not only tools for communication, but also represent the collaboration subject space in 3D. This eases understanding of spatial situations for collaborators by reducing a user's mental effort needed to create an insight of the geographic space visualized by 3D city model [2]. Hence, it is easier for a user to recognize a real-world spatial situation in 3D GeoVEs compared to 2D maps [3].

The contribution of this paper is a model for synchronous and asynchronous communication in 3D GeoVEs such as virtual 3D city models. We present communication tools that serve in combination with a standard-based persistence model for geospatial annotations as basis for collaboration concerning geospatial subjects. The two are combined to facilitate collaboration within 3D GeoVEs. Additionally we propose an approach for integrating complex structured, georeferenced information into 3D representation to avoid context switches. These would be required if such information is presented outside the 3D GeoVE.

The remainder of this paper is organized as follows: Section 2 describes related work, while Section 3 explains communication means for collaboration in 3D GeoVEs. Section 4 subsequently introduces our concept for information storage, during a collaboration process, followed by Section 5 illustrating our approach for visualizing complex structured information within a 3D GeoVE. Section 6 describes a prototypical implementation of our collaboration model. In Section 7 the advantages and drawbacks of the approach are discussed. Finally, future research directions are outlined in Section 8.

2 Related Work

Collaboration in 3D virtual environments has been a topic for more than 20 years [4,5,6,7,8]. Applications for *Computer Supported Collaborative Work* are classified by their distribution of the collaboration process in space (local vs. distributed) and time (synchronous vs. asynchronous) [9,10]. Synchronous collaboration demands for interaction means as well as an awareness for participating collaborators and the context of a situation [7]. Additionally, asynchronous collaboration processes need to effectively model, describe, and store the information created in course of a collaboration. Several researchers have done work in the area of modeling geospatial annotations [11,12,13]. While many approaches use a single point location for georeferencing information, spatial references of information is often more complex. This demands for a more general model of an annotation's spatial reference to store . The system introduced by Rinner et al. [14] provides such a model for collaboration using georeferenced arguments in discussions, but relies on a direct database access, which makes it hard to reuse the data efficiently. Furthermore, many approaches are limited to 2D reference

geometries - nevertheless sufficient for their purpose since they are based on maps as collaboration tools [15,16,17]. Our collaboration approach targets 3D GeoVEs. An extended *Geographic Markup Language* (GML) [18] based model for annotations together with separately modeled spatial references [19] is used for handling of annotations and their spatial references.

The management of annotations in a virtual environment is a complex task [20], especially if they are embedded into the 3D environment. There are multiple 3D annotation techniques that use simple labels, like short texts or icons, to markup entities or geometries [21,22,23,24]. Said simple annotation contents are not fully sufficient for visualization of complex structured data, i.e., annotations equipped with metadata, which additionally provide interactive features for data exploration. Previous attempts dealing this problem integrate visualization approaches known from 2D user interfaces into 3D virtual environments [25]. Andujar et al. [26] employ Qt[1] widgets rendered as textures to enable a fast and easy creation of virtual data representations. They these widgets to control the virtual environment and to explore complex data. Since the functionality of the Qt user-interface library has grown in recent years, the framework offers a wide range of possibilities for data display and interaction. Our system provides interactive visualization of Qt widgets embedded into the virtual environment, which enables structured display and exploration of data associated with a spatial subject.

Jung et al. [27] performed a user study to evaluate an asynchronous collaborative virtual environment for architectural design. Besides textual annotations associated with points in 3D space, the users demanded a way to express change requests or ideas visually. Commonly Sketches have been described as an efficient tool to communicate ideas, opinions, and proposals. Several approaches already use sketches to convey visual or spatial information in 3D environments [28,29]. Heer et al. [30] analyzed the collaborative annotation of data visualizations. In their user study, they identified sketch drawing on those visualizations as an expressive means. Especially pointing in sketches was used frequently. Our system makes use of sketches as communication means. They can be drawn collaboratively during a synchronous collaboration and also be stored alongside with metadata for asynchronous collaboration.

Large, interactive, and heavily distributed virtual environments, such as *Second Life*[2] or *Twinity*[3] support immediate interaction throughout large numbers of participants, embodied by avatars. Their gesticulation are used for pointing and expressing a variety of feelings. Verbal communication means are integrated into clients. Many concepts in Second Life, i.e., signs for information display, are real-world metaphors. By mapping geographical regions onto parts of the Second Life world it is also used for geocollaboration [31]. In contrast to these online virtual worlds we do not store the state of a virtual environment, but only data that is explicitly designated to be persistent.

[1] http://qt.nokia.com
[2] http://www.secondlife.com
[3] http://www.twinity.com

3 Communication Means for Collaboration within 3D City Models

Communication is the central part of a collaborative process. Therefore, communication means are needed that assure efficient transfer of information such as ideas, opinions, options, or proposals efficiently from one user to other collaborators. Due to the spatial nature of the collaboration subject in 3D GeoVEs, large parts of the communicated information is of visual, respectively spatial nature. Standard communication means such as text or audio chats are less suitable to convey such information. They imply a conversion of the form of information from a visual (spatial) form of information to a non-visual form like spoken or written text. In each conversion, information contained in spatial concepts may be lost [32]. Therefore, tools should support users to express the visual part of their ideas efficiently (Fig. 1). The most important ones are markups for 2D and 3D geometries (points, areas, and volumes), geographic features, or groups of those. They facilitate making precise geographic references and, therefore, avoid disambiguities in communication. Verbal communication on the other hand are used to convey non spatial, domain-specific information. We assume that usually external tools for telephony or chat are used for this kind of communication.

In 3D virtual environments, a user's current view is an important factor: information about visible features, current activities, or intents are at least partially encoded. Because of this, a collaborative environment should support both, independent and shared scene views [33]. To share a user's view during synchronous collaboration sessions virtual cameras of all participants need to be synchronized. Further, the user that controls the virtual camera movement has to be determined. We implemented a broadcast-subscription mechanism to enable multiple distinct camera streams with different users subscribed to them in parallel. Through sharing the camera parameters only, users are still able to adapt their model visualization style and integrate different kinds of domain

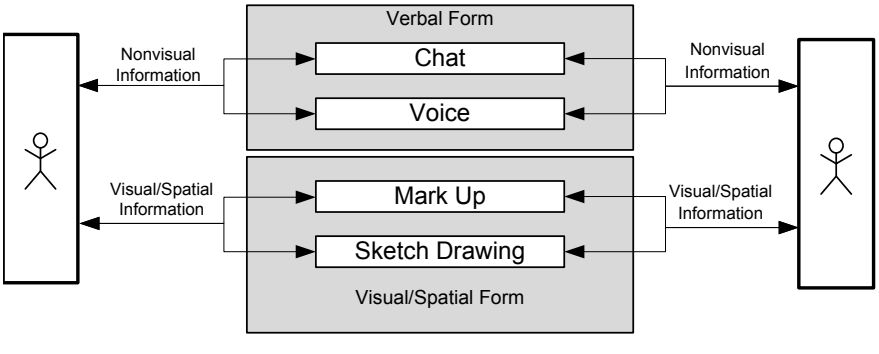


Fig. 1. Communication means for collaboration within 3D GeoVEs. Because conversions between visual information and non-visual form is lossy, visual information is communicated most efficiently using a visual form.

specific data. We are thus using a relaxed variant of the "What You See is What I See" (WYSIWIS) concept [34]. If collaboration is performed asynchronously, a user's view of the scene should still be available to reproduce his spatial context.

To communicate space-related proposals or ideas, e.g., variants of arrangements of geographic features, design options, or routes, they have to be described effectively to minimize loss of information. A 3D scene can be annotated by georeferenced 3D geometry or free-hand sketches to encode this visual information in a visual form (Fig. 1). Additional 3D geometry, e.g., in form of modeled variants of a building, can convey more details of a design or proposal then a hand-draw sketch. Said annotation geometry more difficult to create and adapt compared to sketch annotations, particularly if proposals should be communicated [27]. Moreover, sketch-based depictions are found useful for visualizing ideas that are in early stages of development or focussing certain aspects of a design or proposal [35]. "Sketches stimulate viewers more than shaded images to discuss and actively participate in design development" [36]. So the sketchiness and unfinished look of drawings encourage people to critically scrutinize ideas expressed using a sketch. Their informal character and easy creation make them an essential part of our communication model for synchronous and asynchronous collaboration. Sketching is an effective form of annotation if the 3D GeoVE application is used with touch input devices such as a tablet PC, smart phone, or a tangible wall-mounted display, since those devices provide a more natural interface for drawing.

Free hand sketches can be connected to objects or scene views. While object-based sketches require 3D geometry as surface to draw onto, for view dependent sketches this surface is defined implicitly as the virtual camera's projection plane. Thus, view dependent sketches are more suitable to roughly describe spatial issues,

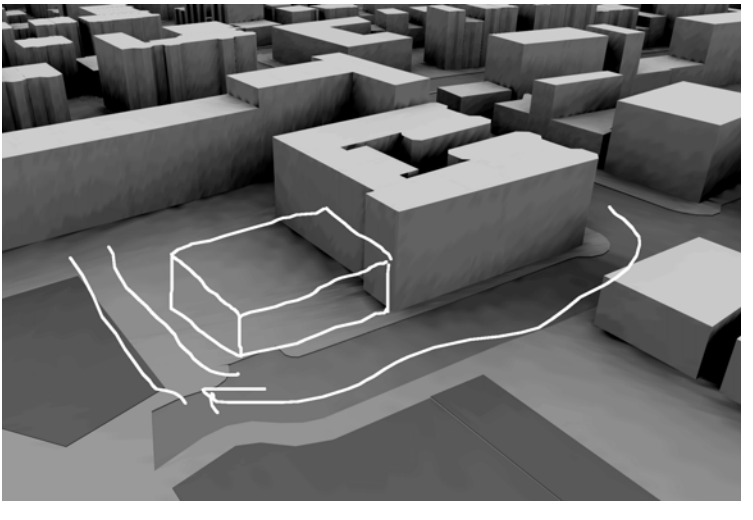


Fig. 2. Example for a sketch illustrating a building extension and its road access

e.g., in early stages where no geometry is available. Further, view-based sketches can be created more easily since users do not need to handle the complexity of the third dimension for drawing. We therefore decided to use camera position dependent sketches drawn onto the scene depiction. Fig. 2 provides an example for such a sketch that describes a proposal for which no 3D geometry exists.

To communicate the different parts of space-related problems in a way that minimizes loss of information, means to communicate spatial and non spatial information are combined.

4 Documentation of Collaboration Processes

Collaboration aims at solving apparent problems when information is shared between co-workers, i.e., knowledge of domain experts in form of opinions, ideas, or proposals. Geospatial annotations are collaboration artifacts that have to be stored to allow comprehension of the collaboration process. For spatial collaboration subjects, such as an urban planning project, information and its spatial references have to be connected in a way that they both can be evaluated and understood later on. An annotation's spatial references are described, encoded, and stored precisely to avoid a loss of information. They associate annotations with geospatial features, 3D geometry, or view description in real world coordinates. Besides the spatial subject of an annotation, information about the context in which an annotation was created helps to comprehend a user's statement later on. Thus, additional metainformation, such as camera parameters, author information, creation time as well as information about the current collaboration session is stored and linked to its respective annotation.

Data created in course of collaborative work should be applicable throughout different specialized software systems that are used by domain experts, e.g., urban planners, architects, or security specialists. To ensure interoperability and to allow integration of collaboration data into such systems, we use a transactional Web Feature Service (WFS-T) [37], which is a standard interface for serving and storing geodata. A GML-based WFS application schema defines the encoding of information created during a collaboration process [19], such as session and user information, geospatial annotations and their precise 3D spatial references. These are modeled as distinct features to ease explicit sharing of instances, i.e., multiple annotation objects sharing spatial references.

The usage of a WFS for data management enables a broad range of applications to integrate collaboration data. For example, GIS-tools can be used for collaboration-data analysis. Besides the obligatory GML output, multiple other output formats are supported. We use XSL-transformations to convert results into KML documents and thus simplify integration into KML-enabled clients such as Google Earth[4], Nasa World Wind[5], and Bing Maps 3D[6].

[4] http://earth.google.com/
[5] http://worldwind.arc.nasa.gov/
[6] http://www.bing.com/maps

5 Embedding Interactive Data into 3D Environments

Information associated with spatial references can be externally visualized using 2D widgets provided by user interface libraries (e.g., Qt, GTK[7]). Alternatively information can be internally visualized by embedding those widgets into virtual 3D space. External visualization uses separate windows and, therefore, spatial references must be encoded into the 3D visualization. The internal visualization, however, implicitly encodes spatial references within the 3D virtual world.

Our client application uses embedded widgets for complex structured data (annotations and their metainformation listings or forms) (Fig. 3) to allow users a seamless interaction with the 3D GeoVE . Users can interact with the widgets to control the system, e.g., taking the camera position of an annotation's author or highlighting all spatial references when clicking a button using the mouse pointer. Systems without window management, e.g., CAVE [38] or a kiosk systems, could profit from this kind of embedded interface [26]. In addition, application development based on Qt widgets is very well supported by tools. This helps to create such widget-based interactive 3D user interfaces quickly.

Embedded widgets are integrated into the 3D space of a 3D GeoVE. Especially when collaborators share the same scene view, 3D widgets can be included into the collaboration process and, therefore, serve as collaboration subjects itself.

Fig. 3. Example of a widget embedded into a 3D GeoVE. The widget shows a website and an annotation's metainformation. In the background other annotations are displayed in different, distance-dependent representations.

[7] http://www.gtk.org

Users are able to talk about and to annotate the visualization of the collaboration data itself like any other scene view using free-hand sketches or markup elements, such as arrows for pointing.

6 System Implementation

We implemented a prototypic system that supports synchronous and asynchronous collaboration based on 3D city models. Our architecture is divided into three components (Fig. 4):

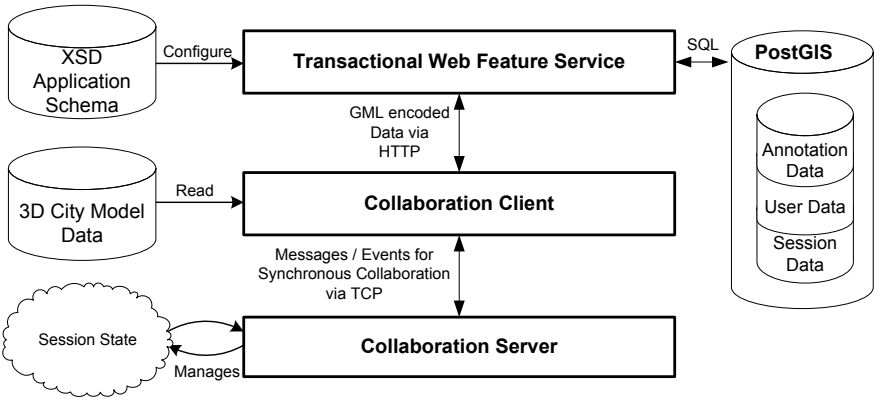


Fig. 4. Architecture of our system for geocollaboration. All persistent data is stored by the WFS-T service in a PostGIS database. The Collaboration Server manages TCP-based message passing and the state of a synchronous collaboration session. 3D visualization and interaction is handled by the collaboration client application.

- A C++ client for visualization and user interaction
- A collaboration server handling synchronous collaboration
- A transactional WFS implementation encapsulating storage and lookup of collaboration data

The encoding format for collaboration information that has to be stored is described using a GML-based WFS-application schema [19]. It defines feature types as well as their relations. As service implementation we use the open source implementation from the deegree[8] project. A PostGIS[9] database is used as data-backend for the service. For generation of KML-encoded output the service is configured using XSL-transformations. This way, collaboration data can be integrated into KML enabled clients for exploration and analysis.

The collaboration server for handling of volatile session data, such as the currently participating users and their positions in the 3D GeoVE, is implemented

[8] http://www.deegree.org
[9] http://postgis.refractions.net

in C++. We use a slim message passing protocol via TCP for communication between server and client applications. Messages are used to transmit a variety of data types, i.e., line strings of sketches, camera positions and orientations for camera synchronization, or text messages for the chat implementation.

The system's user interface is implemented through the Collaboration Client component. Each user executes an instance of the client application. This application visualizes a virtual 3D city model interactively. The distribution of the model is done using an XML descriptor file, which specifies access to a file containing model data. While other distribution methods, such as loading the model through a CityGML [39] serving WFS, are possible, we decided the simplest, file-based one was sufficients for our needs. Georeferenced images (e.g., satellite images or rendered maps) can be included as terrain textures to improve the visual quality of the rendering and provide additional orientation to the user. The client is implemented using the Qt user-interface framework and OpenGL as rendering back-end. Communication with the WFS relies on the Qt implementation of the HTTP-protocol, which is used to manage network communications with the service.

A user may perform sketch drawings on the view plane. Since the current view plane is defined by the camera position and alignment, those drawings are view dependent. Hence, during synchronous collaboration they are only allowed when users share the same scene view. To store those sketches the line strings of a sketch drawing are inversely projected into the 3D geographic coordinate system and encoded as GML geometry.

To optimize information display inside the virtual environment our approach also addresses problems like visual clutter and decreasing screen space with increasing camera distance in 3D virtual environments. To adapt the display of information in 3D, we use an visualization that adjusts itself with regard to the distance from the location of the virtual camera. Annotations for the same spatial reference are grouped to reduce the total number of elements displayed at once in the 3D GeoVE. We distinguish three levels of semantic zoom [40] for information integration into 3D GeoVEs (Fig. 3):

Level 0 - Far distance. An icon, which symbolizes the existence of an information associated with this region.

Level 1 - Medium distance. A group of icons. Each icon symbolizes a category of information, e.g., information, hint or question. Together with the icon the number of information units (annotations) is displayed. The categories are chosen exemplarily.

Level 2 - Near distance. An embedded Qt widget (Fig. 3) showing the information connected to the spatial subject and providing interaction possibilities.

We use the Qt widget engine for embedding an interactive widget into the 3D GeoVE. This user-interface framework provides an OpenGL framebuffer object as paint device, which enables rendering widgets directly into an OpenGL texture (Fig. 5). Rendering a widget as texture is very efficient because no explicit data transfer from the memory of the graphic hardware is necessary. In this way, refresh rates can be achieved that are sufficient to enable video playback

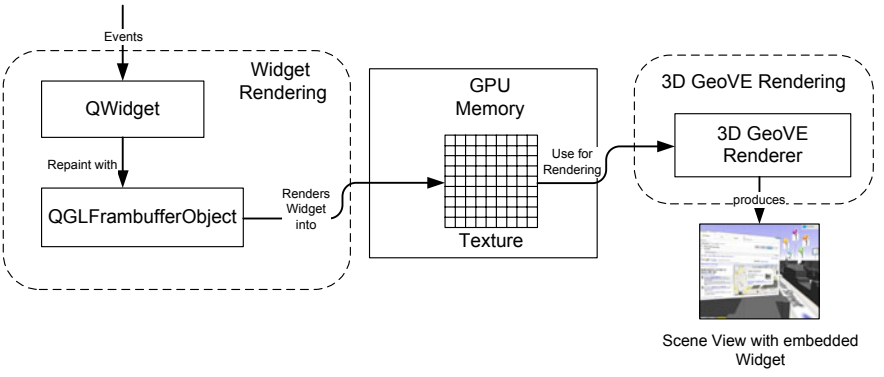


Fig. 5. Process of efficiently rendering an embedded widget. A QGLFramebufferObject is used as rendering device for the Qt widget. An OpenGL texture in GPU memory is used as render target. No separate transfer of pixel data to the GPU for rendering of the 3D GeoVE is necessary.

using such widgets. A single quad, which is placed above the spatial reference of the annotation, is used as underlying geometry for texturing. The connection between annotation content and annotated spatial geometry is strengthened by this annotation placement strategy.

Besides widget display, input events must be mapped to the virtual widget. First of all, for each widget a test is necessary to determine whether a mouse event happened above the depiction of the 3D embedded widget. Through saving the projection, orientation, and model-view matrix for each embedded widget when the OpenGL rendering is performed, this test is performed efficiently. The mouse position is transformed into widget coordinates using these matrices. If these widget coordinates comply with the encapsulated Qt widget's boundaries the event is forwarded. Afterwards the widget texture is updated and the 3D scene is redrawn.

7 Discussion

Data created in course of collaboration sessions can be integrated in a variety of WFS or KML enabled clients. As a test, we used a generic WFS and WMS client (Gaia[10]), which is able to cope with 3D GML-geometries, to create map-based visualizations. We combined the georeferenced collaboration information received from our WFS with several layers originating from diverse WMS services covering the same area (Boston, MA, USA). The result was useful for an overview of annotated regions. After configuring the feature's display and styling the information the data could be usefully combined with other, possible domain-specific data and map layers.

[10] http://www.thecarbonproject.com/

Currently, the display of 3D scene-embedded information is clustered in an straight forward way. Visualized information groups are tightly coupled to the model of spatial references, which describes georeferenced geometry. Information belonging to one spatial reference creates one group, depicted by a single element within the 3D GeoVE. It would be possible to create clusters that use spatial or functional properties as clustering parameters as well as dynamic clustering criteria. Such an approach for integration of information into 3D virtual environments would benefit from the interactivity of 3D virtual environments. For example, view-dependent animations can be used to highlight creation or merge of annotation clusters dependent to the camera position.

At the moment, we are using a simple chat component, which optionally integrates collaboration events, such as annotation creation or joining or leaving users. The log created by our chat component shows a time ordered overview over the collaboration process. This component could be expanded by information about actually visible geographic markups at the time of message sending, which would additionally allow to trace the spatial course of a discussion. This could be used to implement and extend of the concept of "Argumentation Maps" [14] for 3D GeoVEs.

A sketch drawing can be helpful for communicating concepts that are hard to describe verbally. We created several sketch annotations using a mouse as input device and also using a tangible large-scale, wall-mounted display. It turned out that sketch creation using such an tangible display is much more natural than using a standard mouse. In the future, additional tools, e.g., a text typing integration into the sketch editor or prepared shapes, as they are found in conventional 2D drawing programs, could be added to improve the support for sketch creation using standard input devices. Additionally, the advantages and visualization mechanism for object sketching like the one provided by Sin et al. [28] could be further evaluated and integrated.

Synchronous and asynchronous collaboration have different requirements regarding the type of communication. In contrast to asynchronous collabration, for synchronous collaboration, instant communication between collaborators is possible. Hence, asychronous collaboration scenarios need persistent annotations with information about the context of their creation. Because of the other existing communication means in synchronous collaboration, annotations are not the primary communication means but provide a tool for persisting knowledge gained during an synchronous collaboration session.

The basis for synchronous collaboration and integration of annotations into the 3D GeoVE is geographic space and also model data. This eases adaptation and usage of synchronous and asynchronous collaborative tools into different systems. Nevertheless, collboration sessions might need contraints to direct the collboration process towards a specialized problem. Such authoring functionality, e.g., defining boundaries or categories for annotations, is not supported by tools right now. This would be necessary to apply our collaboration approach for more application areas and more specific problems. At the moment such definitions are specified exemplarily.

The current collaboration model is limited regarding its support for group collaboration. So no access registrictions or rights management is implemented by now. Especially large scale applications, e.g., public participation scenarios, could profit from annotations (information as well as sketches). Precise spatial boundaries of reference geometries and metadata collected during annotation creation in the 3d GeoVE allow for detailed analysis of this data. In this case, the informal nature of sketches is a disadvantage. The meaning of a sketch cannot be automatically analyzed, while text analysis techniques can be used for textual annotations.

8 Conclusions

We have shown concepts that support connecting synchronous collaboration (e.g., planning teams, virtual meetings) with asynchronous collaboration processes. We found sketches are useful for collaboration especially when a collaboration system with tangible displays is used to markup or to describe ideas of visual nature, such as arrangements or routes. Our approach for visualizing complex structured information in 3D GeoVEs supports large numbers of annotations to be visualized interactively. By integrating arbitrary Qt widgets into a virtual 3D environment we showed a possibility to create interactive scene elements whose applications are widespread. From integration of custom Qt widgets, videos, up to usable websites a large variety of content can be displayed and used for interaction purposes in 3D.

Especially the annotation function used via an OGC WFS interface is considered valuable to support capturing collaboration or other geo-related data for later evaluation. This evaluation can either be performed collaboratively using our system for data exploration or externally using exisiting GIS software. The standard-based interface to collaboration data supports both.

By now, the system has not been used in a real-world scenario. Additional user tests within a real collaboration scenario will be necessary to generate a more representative number of annotations. Analysis of these collaboration artifacts will yield a more thourough understanding of annotation usage, especially regarding the benefits of different types of annotations.

Future research directions could include the modeling of collaboration processes and their implementation in the system presented in this paper. Those defined processes could be used to generate a more fluid user interaction, by guiding users dependend to collaboration objectives to be achieved. Further, our concepts could be integrated into service-based systems for 3D geovisualization, e.g., into slim, web-based clients [41]. This would lower entry barriers for users and therefore open up a larger user base for a collaborative 3D GeoVE. Through our data-backend is already working with a standard OGC service, integration into a service-based geovisualization landscape should be a realistic scenario.

References

1. Döllner, J., Baumann, K., Buchholz, H.: Virtual 3D City Models as Foundation of Complex Urban Information Spaces. In: CORP, Vienna (2006)
2. Sarjakoski, T.: Networked GIS for Public Participation Emphasis on Utilizing Image Data. Computers, Environment and Urban Systems 22(4), 381–392 (1998)
3. Nurminen, A., Oulasvirta, A.: Designing Interactions for Navigation in 3D Mobile Maps. In: Meng, L., Zipf, A., Winter, S. (eds.) Map-based Mobile Services: Design, Interaction and Usability, pp. 198–224. Springer, London (2008)
4. Greenhalgh, C., Benford, S.: MASSIVE: a collaborative virtual environment for teleconferencing. ACM Transactions on Computer-Human Interaction (TOCHI) 2(3), 239–261 (1995)
5. Pang, A., Wittenbrink, C.: Collaborative 3D Visualization with CSpray. IEEE Computer Graphics and Applications 17(2), 32–41 (1997)
6. Kay, A., Smith, D., Raab, A., Reed, D.: Croquet - A Collaboration System Architecture. In: Proc. First Conf. on Creating, Connecting and Collaborating Through Computing, pp. 2–9. IEEE, Los Alamitos (2003)
7. Dodds, T., Ruddle, R.: Using teleporting, awareness and multiple views to improve teamwork in collaborative virtual environments. In: Proc. of the 14th eurographics Symp. on virtual environments (EGVE 2008), Eindhoven, Eurographics Association, pp. 81–88 (2008)
8. Hindmarsh, J., Fraser, M., Heath, C., Benford, S., Greenhalgh, C.: Object-focused Interaction in Collaborative Virtual Environments. ACM Transactions on Computer-Human Interaction (TOCHI) 7(4), 477–509 (2000)
9. Applegate, L.M.: Technology Support for Cooperative Work: A Framework for Studying Introduction and Assimilation in Organization. Journal of Organizational Computing 1, 11–39 (1991)
10. Duce, D.A., Brodlie, K.W., Gallop, J.R., Walton, J.P.R.B., Wood, J.D.: Distributed and Collaborative Visualization. Computer Graphics Forum 23(2), 223–251 (2004)
11. Knapp, S., Coors, V.: The Use of EParticipation Systems in Public Participation: The VEPs Example. In: Urban and Regional Data Management, pp. 93–104. Taylor & Francis, London (2008)
12. Hopfer, S., MacEachren, A.M.: Leveraging the Potential of Geospatial Annotations for Collaboration: A Communication Theory Perspective. Int. Journal of Geographical Information Science 21(8), 921–934 (2007)
13. Schill, C., Koch, B., Bogdahn, J., Coors, V.: Public Participation Comment Markup Language and WFS 1.1. In: Urban and Regional Data Management, pp. 85–92. Taylor & Francis, London (2008)
14. Kessler, C., Rinner, C., Raubal, M.: An Argumentation Map Prototype to Support Decision-Making in Spatial Planning. In: AGILE 2005 - 8th Conf. on Geographic Information Science, pp. 135–142. Springer, Heidelberg (2005)
15. Yu, B., Cai, G.: Facilitating Participatory Decision-Making in Local Communities through Map-based Online Discussion. In: Proc. of the 4th Int. Conf. on Communities and technologies, pp. 215–224. ACM Press, New York (2009)
16. Mittlboeck, M., Resch, B., Eibl, C.: geOpinion: Interaktives geo-Collaboration Framework - 3D-Visualisierung in Google Earth mit OGC WMS- und WFS- Diensten. In: Strobl, J., Griesebner, B.T.,, G. (eds.) Angewandte Geoinformatik 2006, Beiträge zum 18. AGIT-Symp. Salzbur, pp. 464–469. Herbert Wichman Verlag, Heidelberg (2006)

17. Cai, G.: Extending Distributed GIS to Support Geo-Collaborative Crisis Management. Geographic Information Sciences 11(1), 4–14 (2005)
18. Portele, C.: OpenGIS Geography Markup Language (GML) Encoding Standard (July 2007), http://www.opengeospatial.org/standards/gml
19. Klimke, J., Döllner, J.: Geospatial Annotations for 3D Environments and their WFS-based Implementation. In: Painho, M., Santos, M.Y., Pundt, H. (eds.) Geospatial Thinking. Lecture Notes in Geoinformation and Cartography, pp. 379–397. Springer, Heidelberg (2010)
20. Lenne, D., Thouvenin, I., Aubry, S.: Supporting Design with 3D-Annotations in a Collaborative Virtual Environment. Research in Engineering Design 20(3), 149–155 (2009)
21. Maass, S., Döllner, J.: Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes. In: Butz, A., Fisher, B., Krüger, A., Olivier, P. (eds.) SG 2006. LNCS, vol. 4073, pp. 1–12. Springer, Heidelberg (2006)
22. Maaß, S., Döllner, J.: Dynamic Annotation of Interactive Environments Using Object-Integrated Billboards. In: 14-th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG, pp. 327–334 (2006)
23. Ellis, S.E., Groth, D.P.: A Collaborative Annotation System for Data Visualization. In: AVI 2004: Proc. of the working Conf. on Advanced Visual Interfaces, pp. 411–414. ACM, New York (2004)
24. Kadobayashi, R., Lombardi, J., McCahill, M., Stearns, H., Tanaka, K., Kay, A.: Annotation Authoring in Collaborative 3D Virtual Environments. In: Proc. of the 2005 Int. Conf. on Augmented Tele-Existence, pp. 255–256. ACM, New York (2005)
25. Kadobayashi, R., Lombardi, J., McCahill, M., Stearns, H., Tanaka, K., Kay, A.: 3D Model Annotation from Multiple Viewpoints for Croquet. In: 4th Int. Conf. on Creating, Connecting and Collaborating through Computing (C5 2006), January 2006, pp. 10–15 (2006)
26. Andujar, C., Fairen, M., Argelaguet, F.: A Cost-effective Approach for Developing Application-control GUIs for Virtual Environments. In: IEEE Symp. on 3D User Interfaces, 3DUI 2006, pp. 45–52. IEEE, Los Alamitos (2006)
27. Jung, T., Gross, M., Do, E.: Annotating and Sketching on 3D Web Models. In: Proc. of the 7th Int. Conf. on Intelligent User Interfaces, vol. 1, pp. 95–102. ACM, New York (2002)
28. Sin, E.J., Choy, Y.C., Lim, S.B.: Content-based Sketch Annotations for Collaboration. In: ACM SIGGRAPH 2006 Research posters on - SIGGRAPH 2006, vol. 21 (2006)
29. Do, E.Y., Gross, M.D.: As if You Were Here - Intelligent Annotation in Space: 3D Sketching as an Interface to Knowledge-Based Design Systems. In: AAAI Fall Symp. - Making Pen-Based Interaction Intelligent and Natural, vol. 1, pp. 55–57. AAAI Press, Menlo Park (2004)
30. Heer, J., Viégas, F.B., Wattenberg, M.: Voyagers and Voyeurs:Supporting Asynchronous Collaborative Visualization. Comm. of the ACM 52(1), 87–97 (2009)
31. Ondrejka, C.: Collapsing Geography (Second Life, Innovation, and the Future of National Power). Innovations: Technology, Governance, Globalization 2(3), 27–54 (2007)
32. Yao, J., Fernando, T., Tawfik, H., Armitage, R., Billing, I.: Towards a collaborative urban planning environment. In: Shen, W.-m., Chao, K.-M., Lin, Z., Barthès, J.-P.A., James, A. (eds.) CSCWD 2005. LNCS, vol. 3865, pp. 554–562. Springer, Heidelberg (2006)

33. Chastine, J., Brooks, J., Owen, G., Harrison, R., Weber, I.: A Collaborative Multi-View Virtual Environment for Molecular Visualization and Modeling. In: Coordinated and Multiple Views in Exploratory Visualization (CMV 2005), pp. 77–84. IEEE, Los Alamitos (2005)
34. Benford, S., Greenhalgh, C., Rodden, T., Pycock, J.: Collaborative Virtual Environments. Comm. of the ACM 44, 79–85 (2001)
35. Strothotte, T., Masuch, M., Isenberg, T.: Visualizing Knowledge about Virtual Reconstructions of Ancient Architecture. In: Proc. of the Int. Conf. on Computer Graphics, pp. 36–43. IEEE Comput. Soc., Los Alamitos (1999)
36. Schumann, J., Strothotte, T., Laser, S., Raab, A.: Assessing the Effect of Non-Photorealistic Rendered Images in CAD. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems: Common Ground, pp. 35–41. ACM, New York (1996)
37. Vretanos, P.A.: OpenGIS Web Feature Service (WFS) Implementation Specification (Mai 2005)
38. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., Hart, J.C.: The CAVE: Audio Visual Experience Automatic Virtual Environment. Comm. of the ACM 35(6), 64–72 (1992)
39. Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C.: OpenGIS City Geography Markup Language (CityGML) Encoding Standard Version 1.0.0 (2008)
40. Perlin, K., Fox, D.: Pad: An Alternative Approach to the Computer Interface. In: Proc. of the 20th Ann. Conf. on Computer Graphics and Interactive Techniques, pp. 57–64. ACM, New York (1993)
41. Hagedorn, B., Hildebrandt, D., Döllner, J.: Towards Advanced and Interactive Web Perspective View Services. In: Developments in 3D Geo-Information Sciences, pp. 33–51. Springer, Heidelberg (2009)

Cognitive Invariants of Geographic Event Conceptualization: What Matters and What Refines?

Alexander Klippel[1], Rui Li[1], Frank Hardisty[1], and Chris Weaver[2]

[1] Department of Geography, GeoVISTA Center
The Pennsylvania State University, PA, USA
{klippel,rui.li,hardisty}@psu.edu
[2] School of Computer Science and Center for Spatial Analysis
The University of Oklahoma, OK, USA
weaver@bachman.cs.ou.edu

Abstract. Behavioral experiments addressing the conceptualization of geographic events are few and far between. Our research seeks to address this deficiency by developing an experimental framework on the conceptualization of movement patterns. In this paper, we report on a critical experiment that is designed to shed light on the question of cognitively salient invariants in such conceptualization. Invariants have been identified as being critical to human information processing, particularly for the processing of dynamic information. In our experiment, we systematically address cognitive invariants of one class of geographic events: single entity movement patterns. To this end, we designed 72 animated icons that depict the movement patterns of hurricanes around two invariants: size difference and topological equivalence class movement patterns endpoints. While the endpoint hypothesis, put forth by Regier (2007), claims a particular focus of human cognition to ending relations of events, other research suggests that simplicity principles guide categorization and, additionally, that static information is easier to process than dynamic information. Our experiments show a clear picture: Size matters. Nonetheless, we also find categorization behaviors consistent with experiments in both the spatial and temporal domain, namely that topology refines these behaviors and that topological equivalence classes are categorized consistently. These results are critical stepping-stones in validating spatial formalism from a cognitive perspective and cognitively grounding work on ontologies.

Keywords: Geographic event conceptualization, topology, similarity, spatial cognition.

1 Introduction

The world is dynamic. The embodied human mind possesses evolved cognitive mechanisms that allow it to effectively process dynamic information. We can, for example, maintain a consistent identification of objects even though they change during optical flow (Gibson, 1979), or segment continuous information into meaningful units (e.g., Tversky, Zacks, & Hard, 2008). Without the capacity to make sense of dynamic information, humans would be unable to survive.

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 130–144, 2010.

Recent progress in computer technology allows us to tailor behavioral experiments to deliver results that help to create a better understanding of how the human mind understands dynamic information (events). Shipley (2008) writes: "The advent of computers allows us to control events with greater flexibility than ever before. It is time to seriously consider the appropriate place for events in our science. At the risk of being overly polemical, events appear to be a fundamental unit of experience, perhaps even the atoms of consciousness, and thus should be the natural unit of analysis for most psychological domains." (p. 5)

We tailor our experiments to establish frameworks of movement/event characterization with the goal of improving the cognitive adequacy (Strube, 1991) of spatial formalisms (e.g. Freksa, 1991). Formalisms are at the heart of many theories in geographic information science. Behaviorally validating these formalisms has long been recognized as a crucial step to envision the next generation of theories and applications in spatial sciences (Montello, 2009).

We focus in particular on event conceptualization. We define event conceptualization as category construction (Medin, Wattenmaker, & Hampson, 1987), meaning that we are interested in how humans naturally categorize geographic events. We seek to identify, from a cognitive perspective, factors that are used to distinguish various events. More specifically, movement patterns of individual entities. One way to distinguish these movement patterns and create meaningful units is through the identification of transformational or structural invariants (Gibson, 1979; Shaw, McIntyre, & Mace, 1974; Egenhofer & Al-Taha, 1992).

Researchers in most scientific fields have addressed the topic of invariants' importance to the cognitive systems. Klix (1992) refers to Descartes as maybe the first to make this point: „Das Menschliche Denkvermögen bleibt immer ein und dasselbe, wenn es sich auch den verschiedensten Gegenständen zuwendet, und es erfährt durch ihre Verschiedenartigkeit ebenso wenig eine Veränderung wie das Sonnenlicht durch die Mannigfaltigkeit der Gegenstände, die es bestrahlt." [The human mind stays the same even though it may turn to different objects. In this sense it is like sunlight that does not change either although it shines on many different objects.]

While this argument may be a bit too strong, in a similar vein we find researchers such as Robert Shaw (1974) and J. J. Gibson (1979), both of whom make strong arguments for invariants in the perception of objects and events. Shaw refers to properties that do not change as *transformationally invariant*; Gibson refers to a temporally constant characteristic of the environment as a *structural invariant* (see also Shipley, 2008). The importance of identifying invariants of events is also noted by Galton (2000), who speaks of our ability to intersubjectively identify invariants of time that allow us to construct a shared understanding of our physical (and social) environments. Without this agreement on certain characteristics of spatial environments that ground our meaningful understanding of spatial environments (Scheider, Janowicz, & Kuhn, 2009), the concept of a shared reality and our ability to communicate about this reality would not be possible.

But how can we characterize invariants in dynamic spatial environments? In the qualitative research community, topology has been identified as a way to define invariants (topological equivalence) in order to characterize spatial information in such a way that it becomes possible to model commonsense representations and reasoning. Topology, unquestionably, is playing a central role in characterizing movement

patterns and in bridging the (semantic) gap between a formal characterization of movement patterns and the human understanding of movement patterns (e.g., Kurata & Egenhofer, 2009). Numerous research papers address the characterization of the movement patterns of particularly individual agents (whether they are people, vehicles, or hurricanes) using topological characterization (e.g., Stewart Hornsby & Li, 2009). The motivation to focus on topological characterization stems from the importance that topology plays in a) efficiently representing spatial information, b) in the cognitive understanding of spatial environments, and c) linguistic distinction of spatial relations that are reflected by topological equivalence classes (Cohn, 1997). Yet, while static spatial relations have long been of interest and have been addressed from both the formal and the cognitive behavioral side, movement patterns (events)—while being the focus of formal characterization—have not seen the same attention from a cognitive behavioral perspective. Shipley, as mentioned above, points out that the advent of computers allows us now to control events and event characteristics with unprecedented detail. This control is necessary to deepen our understanding of the cognitive processes underlying the perception and cognition of events.

The reminder of this article is structured as follows. We start by describing an experiment designed to answer a crucial question about the dominance of potentially competing invariants, namely the effects of size differences and topologically defined ending relations of movement patterns. In the discussion and conclusion section, we show how the results of the experiment advance knowledge on cognitive conceptualization of movement patterns, how they help to close the semantic gap between formalism and a cognitive understanding of events, and how they can be applied to increase the cognitive adequacy of similarity ratings.

2 Event Experiments

In our previous experiments (Klippel, Worboys, & Duckham, 2008; Klippel, 2009; Klippel & Li, 2009), we analyzed the role that topology plays in the conceptualization of geographic movement patterns. These experiments differ from those of other researchers, such as the numerous experiments by Mark and Egenhofer (e.g., 1994) or Xu (2007), in that we use animated stimuli. Such ongoing experimental development is considered essential for research on the cognitive understanding of events (Shipley, 2008). While some researchers conducting similar experiments in the static domain have found topology to be the most important invariant for distinguishing spatial relations, our results were different in two important ways. First, in experiments with competing invariants such as size differences or different dynamic characteristics (whether one entity is moving or both), we found that it was necessary to reverse the famous statement by Mark and Egenhofer that *topology matters and metric refines*. We found that other characteristics mattered and that topological distinctions were used as refinements. For example, participants first distinguished between one entity moving or both, and afterwards made some topologically induced distinctions. Second, in experiments in which we employed real world scenarios (e.g., a hurricane crossing a peninsula), participants did use topologically defined ending relations as the main cognitive invariant. Two important aspects are worth keeping in mind, however. First, we did not introduce many alternative factors; it would, for example, not make sense to have the peninsula moving rather than the hurricane. Second, even

though topological distinctions were used as the main criterion for categorizing movement patterns, we found that not all topological relations are equally salient from a cognitive perspective. Most strikingly, we found similarities to work by Lu and Harter (Lu & Harter, 2006), who employed Allen's intervals as a hypothesis for participants and cognitively salient ending relations. Their findings indicate that participants group together temporal relations that show some kind of overlap and distinguish them from those relations that do not overlap.

We set up an experiment to shed more light on the question of the dominance of invariants in constructing categories of geographic events. We are interested in two aspects that result from previous experiments, but have not been answered before together: Are topologically distinguished ending relations the cognitively most salient invariant in the category construction of movement patterns? If not, do we still find the different saliencies for individual topologically distinguished ending relations that we find in the absence of size differences? We thus used a real world scenario—a hurricane crossing a peninsula (Klippel & Li, 2009)—but introduced the previously influential factor of size as an additional factor.

Participants. 20 Penn State undergraduate students took part in the experiment; eight were male, the average age was 20.85. Participants were recruited from both Geography and Information Sciences & Technology. They were reimbursed US$10.

Material. We created 72 animated icons, each 120x120 pixels in size. The icons show a peninsula surrounded by water and a hurricane moving toward the peninsula from the top-right corner. Adobe Flash CS4 was used to create the animations and export them in animated GIF format. The animations were further smoothed and enhanced in appearance using the Easy GIF Animator software. Variation was introduced by randomizing both the start and end coordinates of the hurricanes. Start coordinates are all located near the top-right corner. End coordinates were chosen according to one of nine topological equivalence classes (see below and Figure 1).

Given the importance of ending relations for the conceptualization and understanding of events (Regier, 1996; Regier & Zheng, 2007), we selected the end coordinates randomly such that they would fall into one of the topological equivalence classes specified by the conceptual neighborhood graph in Figure 1 on the basis of RCC-8 (Randell, Cui, & Cohn, 1992) or the 9-intersection model (Egenhofer & Franzosa, 1991). Hence, the following ending relations were realized (see also Figure 1): DC1 – the hurricane does not make landfall; EC1 – the hurricane kind of bumps into the peninsula; PO1 – the hurricane just reaches land such that half of the hurricane is on land and the other half is over water; TPP1 – the hurricane makes landfall but is still 'connected' to the water; NTPP – the hurricane makes landfall and is completely over land; TPP2 – same as TPP1 but the hurricane nearly made it out to the water again; PO2 – same as PO1 but on the other side of the peninsula; EC2 – same as EC1 but on the other side of the Peninsula; DC2 – same as DC1 but has crossed the peninsula completely.

For each of these topological equivalence classes, we created eight animated icons that differed in the actual starting and ending coordinates but not the topologically characterized path of the movement through the conceptual neighborhood. Of these eight animations, four showed a large hurricane and four showed a small hurricane. Adding this latter aspect extends previous research in order to answer the question of dominant cognitive invariants in the conceptualization of movement patterns.

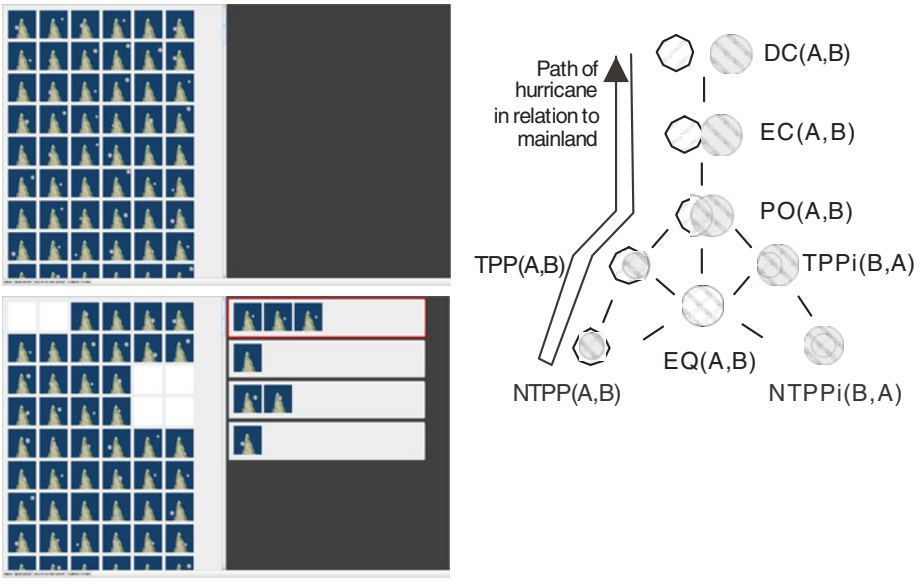


Fig. 1. Screenshot of the experiment interface. Top: Initial screen; bottom: mimicked ongoing experiment. On the right side a conceptual neighborhood graph is depicted.

Procedure. The experiments took place as a group experiment. For this purpose, we equipped a GIS lab in the Department of Geography at Penn State with view blocks such that up to 16 participants could partake in the experiment at the same time but not see each other's screens. The lab is equipped with 24'' Dell wide screen monitors providing excellent conditions for performing a grouping experiment on a screen. The software we used in previous experiments was improved such that at the beginning of the main experiment all animated icons would appear on the left side of the screen (all at the same time) but that moving one icon from the left side into a group (category) on the right side would leave the spot empty (see Figure 1). Otherwise the experiment followed established and tested experimental paradigms.

The participants were randomly assigned to computers, provided consent, and entered anonymous personal data such as age and field of study. They received a short introduction detailing the scenario and the course of the experiment. This text explicitly referred to the stimulus as hurricane and peninsula. It also made clear that there are no right or wrong ways to creates group (categories), and that it is up to the participants themselves to select criteria and the number of appropriate groups (categories). To make this point even clearer, participants were provided with a warm up task and were asked to group animals such as dogs, cats, and camels. The grouping software does not allow for ending a task before all icons are placed into groups, whether in the warm up task or in the main experiment.

In the main experiment, participants were presented with the 72 animated icons. In the first part of the main experiments they created as many groups (categories) as they deemed appropriate. Participants had to explicitly create all groups, that is, there are no groups depicted or implied on the right side of the experimental software user interface (see Figure 1). Participants can create and delete groups, move animated

icons into and out of groups, or move icons between groups. This procedure is referred to alternately as category construction (Medin et al., 1987), free classification, or unsupervised learning (Pothos & Chater, 2002).

After placing all icons into groups, participants entered the second part of the main experiment. In this part, participants were presented again with the groups that they had created in the first part. The groups were presented one at the time, and the participants asked to linguistically label them and to draw a sketch map, a graphic symbol that represents the group. The linguistic labeling task has two parts: to provide a short description of no more than 5 words, and to provide a more detailed explanation of the rationale upon which a particular group was created.

3 Results

We first analyze the category construction behavior of the participants. To this end, we first calculate individual grouping matrices for each participant. Individual matrices encode the grouping behavior for all possible (72 x 72 = 5184) icon pairings as binary values. A '0' in a matrix indicates two icons are not placed into the same category and a '1' indicates that two icons are placed into the same category. We then calculate an overall similarity matrix (OSM) by summing over all individual matrices. An OSM thus encodes similarities on the basis of grouping behavior: the least similarity, '0', indicates that two icons were never placed into the same group; the highest similarity, '20', indicates that the corresponding pair of icons were placed into the same group by all participants.

The OSM is the basis for both the cluster analysis methods and the multidimensional scaling (MDS) that we discuss in the following sections. We validate cluster results using approved and recommended methods (Kos & Psenicka, 2000; Clatworthy, Buick, Hankins, Weinman, & Horne, 2005): a) the comparison of different clustering methods and b) splitting the participant pool randomly in half and comparing the result (that we will not discuss in detail here but that showed similar results).

We compared three clustering methods: Ward's method (or increase sum of squares), average linkage, and complete linkage. Ward's method is often preferred for its use of a statistical measure, that is, minimizing the increase of the sum of squared differences from the group mean. In contrast, average and complete linkage use arithmetic calculations to analyze the similarity structure. Hence, in case these different methods indicate similar clustering structures such as the inferred number of clusters, they can be used as validation.

Figure 2 shows the results of two (out of three) cluster analyses (as mentioned above). The dendrograms reveal that size is the dominating factor. This is shown in both dendrograms (Ward's method and average linkage); they both show that participants used size to create two distinct groups. While complete linkage (see website) does not show the same bifurcated structure on the basis of size, but size is still clearly used to distinguish (cut through) each topologically defined equivalence class. The saliency of topologically defined ending relations thus functions as a refinement. While size is the primary criterion that participants use to differentiate the hurricane movement patterns, the further differentiation based on topologically defined ending

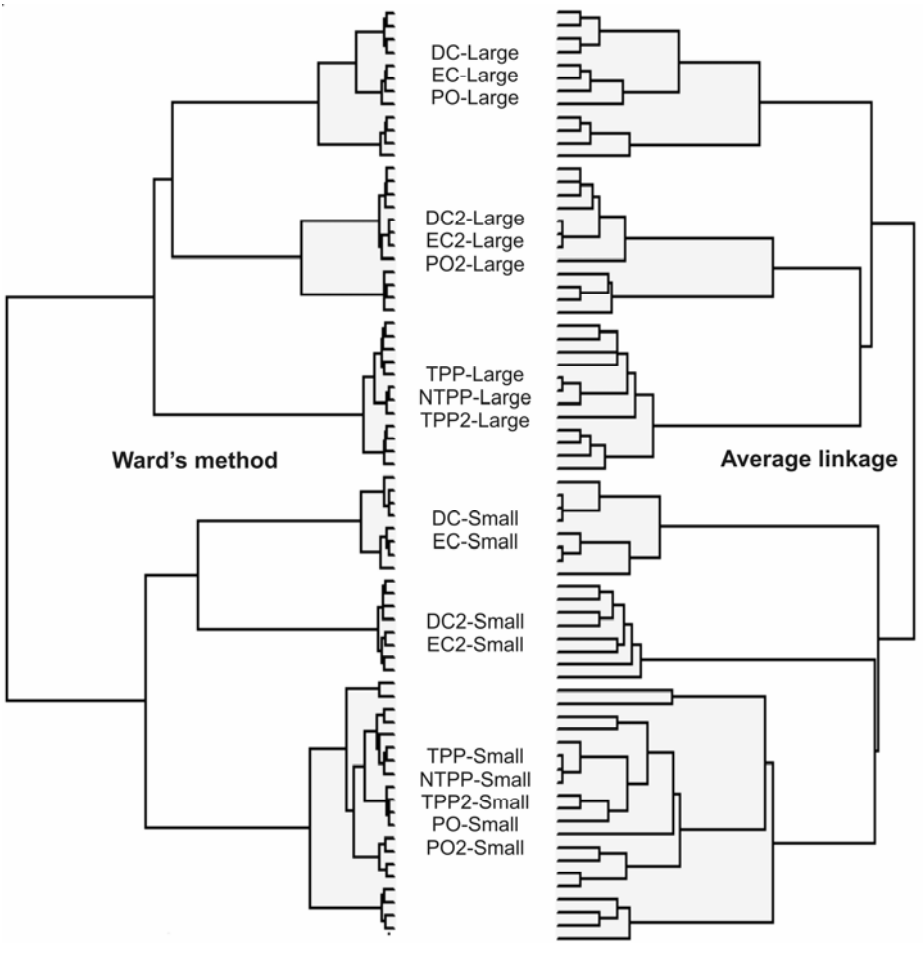


Fig. 2. Depicted are two clustering method: Ward's method and average linkage (the original three cluster analyses can be found at: http://www.cognitivegiscience.psu.edu/cluster-analysis.jpg). The comparison of the different clustering methods shows basically the same results indicating the validity of the interpretation.

relations gives a similar picture to previous experimental results within each size category. Most importantly, we again were not able to replicated findings from static experiments that hint at equal saliency of all topologically defined ending relations (see Knauff, Rauh, & Renz, 1997). In other words, we again find that certain topological relations are conceptually closer together than others, in a pattern that is similar to those of previous experiments (see Klippel & Li, 2009): Ending relations that do not show some kind of overlap (DC and EC) are separated from those that do show some kind of overlap (TPP and NTPP). The exception to this pattern is that the ending relation partial overlap (PO) that has been identified in previous experiments as a "flip-flop" relation is not as clearly conceptualized as other ending relations. The changing character of the PO relation becomes particularly apparent in the case of

small hurricanes. This deviation from the pattern is most likely related to perceptual characteristics, that is, this relation becomes harder to identify properly in cases in which the hurricane is small.

For an additional perspective on the category construction behavior of the participants, we also subjected the OSM directly to a multidimensional scaling algorithm using Clustan™. The results of this analysis are shown in Figure 3. We used a custom-made software tool to visualize MDS results, that is, to place a picture showing only the ending relation of the hurricane movement. The pictures are reduced in size, but with a total of 72 pictures some overlap (which actually indicates high similarity) was unavoidable. One ideal outcome of MDS is to identify axes and label those axes. In our case this is nicely possible. The first axis (see Figure 3) is making the distinction between small and large hurricanes. This axis clearly confirms the analysis that we obtained from the cluster analysis in Figure 2 that size matters. The second axis reflects the conceptual distance between different topologically defined ending relations of movement patterns. This axis reveals that certain topologically defined ending relations—the ones found grouped close together in the dendrograms— are conceptually closer together than others.

To further confirm these results, we used KlipArt (Klippel, Hardisty, & Weaver, 2009) to analyze the category construction behavior together with the linguistic descriptions that participants provided. Figure 4 shows an example of the grouping

Fig. 3. MDS analysis of the category construction behavior. In the ideal case, MDS analysis allows for labeling the axis of the plot. In our case, this can be nicely done as participants clearly distinguished size differences as well as topologically defined ending relations.

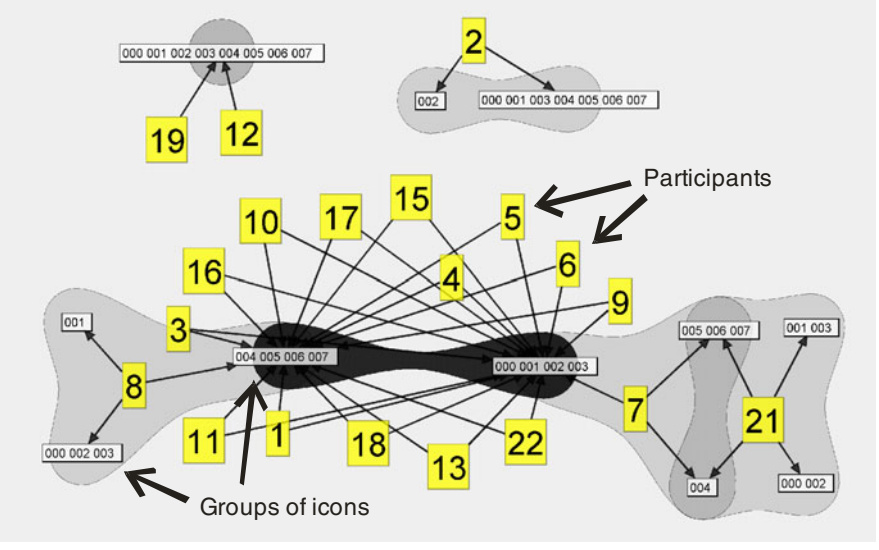


Fig. 4. KlipArt analysis. An in depth analysis reveals the category construction behavior and allows for linking linguistic descriptions with category construction behavior. The Figure clearly shows that, for the movement patterns characterized as DC, most participants (14 out of 20) used size as the main distinguishing criterion (in which icons 000, 001, 002, and 003 show a large hurricane and icons 004, 005, 006, and 007 show a small one). Additional linguistic analysis reveals that 17 out of 20 participants used size in the category construction process.

behavior for hurricane movement patterns that do not make landfall. In this case the conceptual neighborhood path is very short because peninsula and hurricane are always disconnected (DC). It is clear that most participants (14 out of 20) separated the animated icons on the basis of the size of the icons, compared to 2 participants who place all icons in the same category. The linguistic analysis allowed us to shed more light on the rationale of the category construction behavior. It also revealed that participant 7 made a 'mistake' and meant to classify the icons into two categories, participant 2 meant to group all icons into one group, and that participants 8 and 21 did use size as a criterion but made finer distinctions based on where the hurricanes made landfall.

We now turn to the analysis of the linguistic descriptions that the participants created. We focus here primarily on the short descriptions that participants provided for the groups that they created. Participants created a total of 140 groups, hence we had 140 labels with an average length of 3.5 words. The aspect we focused on is the question whether the size of the hurricanes was mentioned either directly, such as '*large* hurricane' or '*small* storm', or indirectly, such as '*weak* hurricane' or '*strong* hurricane'. In 86% of the labels, we find either a direct or indirect reference to the size of the hurricanes.

4 Discussion

The results of our experiment add to the understanding of how events in geographic space are conceptualized. Previous experiments provided insufficient evidence about the

salience of topologically defined ending relations of movement patterns compared to other factors such as size differences. Our experiment closes this gap in our knowledge.

We analyzed different types of data with the intention to cross-validate experimental results. To this end, we collected category construction information (similarity assessments) and linguistic labels. Combining both data sources offers valuable insights into cognitive processes underlying the category construction of geographic events. Additionally, we cross-validated our findings by employing different analysis methods. We used different types of cluster analysis that we compared against one another, performed the same analysis on random subgroups of participants, used multidimensional scaling to improve the interpretation of the data through spatialization (Skupin & Fabrikant, 2007), and additionally, we used a custom-made software tool, KlipArt, that allows us to analyze the category construction behavior of individual or groups of participants for specific groups of icons. Amazingly, all these analyses converge on the same conclusion: *Size matters and topology refines*.

We now have, however, a conundrum: Given the importance of events exhibited through a strong interest of different research communities in spatio-temporal frameworks, it is amazing that a static aspect dominates that category construction behavior or participants. Could it be that size is a domain-specific concept and factor? We could make the argument that size may be more important in one domain than in another. It makes an important conceptual difference, whether a small hurricane passes over a peninsula or whether a large one does. This distinction may not be as crucial for other domains, for example, whether a big tanker crosses a certain area or a small one. However, our previous experiments, that used geometric form (circles) and in which participants were 'only' asked to imagine something geographic, showed a clear dominance of size differences, too. Hence, unquestionably, size differences are an important factor in perception (Wolff, 2008; Lockhead & Pomerantz, 1991) as well as in conception. We would argue that in most cases size (extent) is an conceptually important piece of information.

Despite the dominance of size as the main criterion for categorization, we must point out that topology also plays a role in refining category construction. Even more importantly, we find a pattern that is consistent with those of our own previous experiments (Klippel & Li, 2009) as well as experiments from the temporal domain (Lu & Harter, 2006). This consistent pattern has the following characteristics. Unlike the results of the experiments conducted by Knauff and collaborators (Knauff et al., 1997), topologically defined ending relations of movement patterns do not all have the same conceptual salience. Whereas in the static domain all eight topological relations show comparable salience, in the case of events, there is a clear pattern that distinguishes topological relations that overlap (TPP and NTPP) from those that do not (DC and EC). The one somewhat inconsistent exception is the relation partial overlap (PO).

5 Conclusions

Analyzing movement patterns, developing models for event-oriented approaches to spatial information, and developing a theory of how events at the geographic scale are construed and understood by humans, are all prominent research efforts in geographic

information science. Our research fills a *critical niche*: We are developing a systematic and extensive research framework for the behavioral exploration, assessment, and validation of the semantics of movement patterns. The key aspects of our research are: a) developing a multi-method and multi-data-type analysis framework that makes it possible to open multiple *windows to cognition* and thereby substantially improve our understanding of cognitive processes underlying the conceptualization of events; b) utilizing computational methods to display events as animations to be used in behavioral experiments; c) cross-validating our research findings to improve the reliability of our interpretations; d) tailoring the research design to bridge the semantic gap between formalisms and their capacity to model the human mind.

We demonstrated that, with these prerequisites, the design and evaluation of qualitative formalisms—critical for geographic information science—becomes feasible and can be conducted in great depth. Additionally, we are able to focus on how events at the geographic scale are conceptualized. Step by step we add to a cognitive theory of geographic event conceptualization that is grounded in the theoretical foundations laid out in spatial information science (e.g., Kurata & Egenhofer, 2009; Stewart Hornsby & Li, 2009) but that also allows for evaluating proposed formalisms. We consider this step by step approach essential, as it basically combines the theory building that we find in psychology and cognitive science with topics relevant to spatial information theory. The fruitfulness of this combination has been explicated in the original work by Mark and Egenhofer (e.g., Mark & Egenhofer, 1994), that many researchers, including us, have adopted.

Our future work in this area includes the following directions. While we have shown now that size matters and that topology refines in the conceptualization of hurricane movement patterns, we are aiming for a theory of conceptualizing movement patterns that is able to explain aspects of movement conceptualization that are universal across different domains and those where the semantic context might be influential. The semantic context can be thought of in two ways. The first is that different domains will emphasize or de-emphasize the cognitive saliency of different topological relations. While we do have cross-validation through the work by Lu and Harter (Lu & Harter, 2006) that an important distinction has to be made between topological relations that show some kind of overlap and those that do not, there are the open questions of whether this is the case across different domains (hurricanes, tornados, boats, etc.), and why this does not seem to be the case in the static domain. To this end, we have designed scenarios that are taken from different domains and also added a comparison of statically depicted paths of hurricanes.

The second is that the semantics of a domain might be different for experts and non-experts. This is a crucial topic that has received a lot of attention (Bryant, 2000). There very well may be some fine tuning in category construction behavior when comparing experts and non-experts. We share the belief put forth by Goldstone (1994) that even though we could define arbitrary concepts according to theoretical or contextual information, but humans—experts and non-experts—indeed derive most useful information from perceptual input, and this information is valid even in more complex theoretical constructs.

Two more aspects are interesting to us in this regard. First, how may different topological transformations influence the cognitive saliency of topologically defined ending relations? For instance, if we compare translation (such as hurricane movement) with

scaling (such as the extension and shrinking of a lake), we can create an identical conceptual path through a conceptual neighborhood graph (Egenhofer & Al-Taha, 1992). However, does this lead to a different assessment of the cognitive salience of topologically defined ending relations?

Second, we are working on extending our research framework to evaluate further different formalisms. So far we have focused on conceptual neighborhood graphs based on topological relations between two extended spatial entities. More recently Kurata (Kurata, 2008) proposed an extension of the 9 intersection model, the 9+-intersection model (last year's best GIScience paper).This framework explicitly addresses the modeling of movement patterns by utilizing the source-path-goal characteristics of every movement. Kurata and Egenhofer (2009) use this approach to model the cognitive understanding of movement patterns. The critical question is whether from a cognitive perspective this framework works as is, or whether it requires cognitive adaptations. We are currently working on experiments that will shed light on the question of whether all 26 primitive relations defined by the 9+ intersection model are equally salient from a cognitive perspective.

A second line of research is to use the results of our experiments to inform similarity assessment of movement patterns. One of the most critical aspects in establishing similarity measures is the specification of weights. If we take the conceptual neighborhood graph that provides the basis for our experiments as an example for topological and—more generally—graph-based similarity measures, we find that most research on similarity is using non-weighted edges (Schwering, 2008). If we change, for instance, the underlying formalism from RCC8 to RCC5 (also possible in Egenhofer's models with some differences in the coarser models), then we will also change the similarity between topological relations. Likewise, while the assumption of equal weights does not seem to contradict results using static spatial relations (Knauff et al., 1997), this assumption contradicts findings in the dynamic domain (Lu & Harter, 2006; Klippel & Li, 2009). Hence, it is essential to a) understand the different roles that topology plays in the static and the dynamic domain, and b) to refine topological similarity ratings that are based on graph structures such as the conceptual neighborhood graph. Our results thus far provide input to such refinements and we are currently working on integrating them into the SIM-DL framework (Janowicz, Keßler, Schwarz, Wilkes, Panov, Espeter et al., 2007). The two possibilities we are exploring are the use of fusion coefficients that can be derived from different clustering algorithms, or, alternatively, we are exploring ways to use the similarity ratings directly. In line with the future research directions specified above, we will derive similarity values for different spatial and semantic contexts with the goal of establishing a universal similarity measure that would generally improve similarity assessments. Nonetheless, we continue working on context-specific tailoring of similarity assessments to non-topological information under different contexts and in different semantic domains.

Acknowledgements

We would like to acknowledge Thilo Weigel who implemented the grouping tool that Markus Knauff and collaborators used, which inspired our grouping tool. We sincerely thank Stefan Hansen for implementing our first grouping tool. Research for this

paper is based upon work supported by the National Science Foundation under Grant No. 0924534 and funded by the National Geospatial-Intelligence Agency/NGA through the NGA University Research Initiative Program/NURI program. The views, opinions, and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Science Foundation, the National Geospatial-Intelligence Agency, or the U.S. Government.

References

Bryant, R.: Discovery and decision: Exploring the metaphysics and epistemology of scientific classification. Fairleigh Dickinson University Press Associated Univ. Presses, London (2000)

Clatworthy, J., Buick, D., Hankins, M., Weinman, J., Horne, R.: The use and reporting of cluster analysis in health psychology: A review. British Journal of Health Psychology 10, 329–358 (2005)

Cohn, A.G.: Qualitative Spatial Representation and Reasoning Techniques. In: Brewka, G., Habel, C., Nebel, B. (eds.) KI 1997. LNCS, vol. 1303, pp. 1–30. Springer, Heidelberg (1997)

Egenhofer, M.J., Al-Taha, K.K.: Reasoning about gradual changes of topological relationships. In: Frank, A.U., Campari, I., Formentini, U. (eds.) Theories and methods of spatio-temporal reasoning in geographic space, pp. 196–219. Springer, Berlin (1992)

Egenhofer, M.J., Franzosa, R.D.: Point-set topological spatial relations. International Journal of Geographical Information Systems 5(2), 161–174 (1991)

Freksa, C.: Qualitative spatial reasoning. In: Mark, D.M., Frank, A.U. (eds.) Cognitive and linguistic aspects of geographic space, pp. 361–372. Kluwer, Dodrecht (1991)

Galton, A.: Qualitative spatial change. Spatial information systems. Oxford Univ. Press, Oxford (2000)

Gibson, J.: The ecological approach to visual perception. Houghton Mifflin, Boston (1979)

Goldstone, R.: The role of similarity in categorization: Providing a groundwork. Cognition 52(2), 125–157 (1994)

Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., et al.: Algorithm, implementation and application of the SIM-DL similarity server. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 128–145. Springer, Heidelberg (2007)

Klippel, A., Hardisty, F., Weaver, C.: Star plots: How shape characteristics influence classification tasks. Cartography and Geographic Information Science 36(2), 149–163 (2009)

Klippel, A., Worboys, M., Duckham, M.: Identifying factors of geographic event conceptualisation. International Journal of Geographical Information Science 22(2), 183–204 (2008)

Klippel, A.: Topologically characterized movement patterns: A cognitive assessment. Spatial Cognition and Computation 9(4), 233–261 (2009)

Klippel, A., Li, R.: The endpoint hypothesis: A topological-cognitive assessment of geographic scale movement patterns. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 177–194. Springer, Heidelberg (2009)

Klix, F.: Die Natur des Verstandes. Hogrefe, Göttingen (1992)

Knauff, M., Rauh, R., Renz, J.: A cognitive assessment of topological spatial relations: Results from an empirical investigation. In: Hirtle, S.C., Frank, A.U. (eds.) COSIT 1997. LNCS, vol. 1329, pp. 193–206. Springer, Heidelberg (1997)

Kos, A.J., Psenicka, C.: Measuring cluster similarity across methods. Psychological Reports 86, 858–862 (2000)

Kurata, Y.: The 9+-intersection: A universal framework for modeling topological relations. In: Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 181–198. Springer, Heidelberg (2008)

Kurata, Y., Egenhofer, M.J.: Interpretation of behaviors from a viewpoint of topology. In: Gottfried, B., Aghajan, H. (eds.) Behaviour monitoring and interpretation. Ambient intelligence and smart environments, pp. 75–97. IOS Press, Amsterdam (2009)

Lockhead, G.R., Pomerantz, J.R. (eds.): The perception of structure: Essays in honor of Wendell R. Garner. American Psychological Assoc., Washington (1991)

Lu, S., Harter, D.: The role of overlap and end state in perceiving and remembering events. In: Sun, R. (ed.) The 28th Annual Conference of the Cognitive Science Society, Vancouver, British Columbia, Canada, July 26-29, pp. 1729–1734. Lawrence Erlbaum, Mahwah (2006)

Mark, D.M., Egenhofer, M.J.: Topology of prototypical spatial relations between lines and regions in English and Spanish. In: Proceedings, Auto Carto 12, Charlotte, North Carolina, March 1995, pp. 245–254 (1995)

Mark, D.M., Egenhofer, M.J.: Modeling spatial relations between lines and regions: Combining formal mathematical models and human subject testing. Cartography and Geographic Information Systems 21(3), 195–212 (1994)

Medin, D.L., Wattenmaker, W.D., Hampson, S.E.: Family resemblance, conceptual cohesiveness, and category construction. Cognitive Psychology 19(2), 242–279 (1987)

Montello, D.R.: Cognitive research in GISceince: Recent achievements and future prospects. Geography Compass 3(5), 1824–1840 (2009)

Pothos, E.M., Chater, N.: A simplicity principle in unsupervised human categorization. Cognitive Science 26(3), 303–343 (2002)

Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connections. In: Proceedings 3rd International Conference on Knowledge Representation and Reasoning, pp. 165–176. Morgan Kaufmann, San Francisco (1992)

Regier, T.: The human semantic potential: Spatial language and constraint connectionism. The MIT Press, Cambridge (1996)

Regier, T., Zheng, M.: Attention to endpoints: A cross-linguistic constraint on spatial meaning. Cognitive Science 31(4), 705–719 (2007)

Scheider, S., Janowicz, K., Kuhn, W.: Grounding geographic categories in the meaningful environment. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 69–87. Springer, Heidelberg (2009)

Schwering, A.: Approaches to semantic similarity measurement for geo-spatial data: A survey. Transactions in GIS 12(1), 2–29 (2008)

Shaw, R., McIntyre, M., Mace, W.: The role of symmetry in event perception. In: MacLeod, R.B., Pick, H.L. (eds.) Perception. Essays in honour of James J. Gibson, pp. 276–310. Cornell University Press, Ithaca (1974)

Shipley, T.F.: An invitation to an event. In: Shipley, T.F., Zacks, J.M. (eds.) Understanding events: How humans see, represent, and act on events, pp. 3–30. Oxford University Press, New York (2008)

Skupin, A., Fabrikant, S.I.: Spatialization. In: Wilson, J., Fotheringham, A.S. (eds.) The handbook of geographic information science. Blackwell companions to geography, vol. 7, pp. 61–79. Blackwell, Malden (2007)

Stewart Hornsby, K., Li, N.: Conceptual framework for modeling dynamic paths from natural language expressions. Transactions in GIS 13(s1), 27–45 (2009)

Strube, G.: The Role of Cognitive Science in Knowledge Engineering. In: Schmalhofer, F., Strube, G., Wetter, T. (eds.) GI-Fachtagung 1991. LNCS, vol. 622, pp. 161–174. Springer, Heidelberg (1992)

Tversky, B., Zacks, J.M., Hard, B.M.: The structure of experience. In: Shipley, T.F., Zacks, J.M. (eds.) Understanding events: How humans see, represent, and act on events, pp. 436–464. Oxford University Press, New York (2008)

Wolff, P.: Dynamics and the perception of causal events. In: Shipley, T.F., Zacks, J.M. (eds.) Understanding events: How humans see, represent, and act on events. Oxford University Press, New York (2008)

Xu, J.: Formalizing natural-language spatial relations between linear objects with topological and metric properties. International Journal of Geographical Information Science 21(4), 377–395 (2007)

A Visibility and Spatial Constraint-Based Approach for Geopositioning

Jean-Marie Le Yaouanc, Éric Saux, and Christophe Claramunt

Naval Academy Research Institute,
CC 600, Lanvéoc, 29240 Brest Cedex 9, France
{leyaouanc,saux,claramunt}@ecole-navale.fr

Abstract. Over the past decade, automated systems dedicated to geopositioning have been the object of considerable development. Despite the success of these systems for many applications, they cannot be directly applied to qualitative descriptions of space. The research presented in this paper introduces a visibility and constraint-based approach whose objective is to locate an observer from the verbal description of his/her surroundings. The geopositioning process is formally supported by a constraint-satisfaction algorithm. Preliminary experiments are applied to the description of environmental scenes.

Keywords: Landscape perception, place descriptions, scene-finding approach, geopositioning.

1 Introduction

Geopositioning is a process whose objective is to relate a geographic location to a given entity, activity or person. This is supported by qualitative references to locations we employ in everyday discourse, *e.g.* place-names, and quantitative representations used in many activities based on coordinates-based navigation. Early geopositioning systems have been widely applied to quantitative models and geometrical representations of space. Despite the interest of these approaches for cartographical applications, they do not completely reflect the way a human perceives and describes his/her environment since he/she preferably stores and processes qualitative information. This is particularly relevant for natural environments since they do not have well-defined emerging structures similar to those present in an urban environment. Perception encompasses cognitive principles that favor memorization of the main properties of an environment, and potentially communication of these properties to an external addressee using natural language [1,2]. These descriptions are essentially qualitative and based on common sense, *i.e.*, intuitive concepts we daily manipulate to interact with our environment [3,4,5].

While the interpretation of spatial relations for the location of entities has long been studied, they have been hardly considered for the geopositioning of an observer perceiving his environment. The objective of the research presented in this paper consists in developing a model suitable with perception and spatial

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 145–159, 2010.

cognition, but also appropriated for the processing of quantitative spatial data. We consider the case of an observer located at a fixed vantage in a natural landscape, perceiving his/her 360° surroundings, and who is asked to provide a description of his environment to an external addressee. The description of such an environment underlines the salient entities of space, the spatial relations between them, and the structural properties of the environment. Entities are identified according to a semantic categorization, their proximity and orientation with respect to the observer [6]. The research presented in this paper extends this modeling approach by identifying the possible locations of the observer from the interpretation of the description of his surroundings. It is supported by a visibility and constraint-based approach. This provides a support for search and rescue approaches by bridging the gap between a qualitative map resulting from the direct perception of a scene, and a quantitative representation of the environment given by a GIS database.

The remainder of the paper is organized as follows. Section 2 presents the modeling background of the approach and a conceptual representation of an environmental scene. Section 3 introduces the context of this research and develops the modeling approach, based on the concept of visibility of salient entities and the interpretation of spatial relations as spatial constraints. Lastly, Section 4 draws the conclusions and outlines further work.

2 Modeling Background

In a previous work, we have introduced a structural model of a scene generated from the interpretation of a verbal description. It identifies the salient entities, spatial relations between them, and spatial constructs of the landscape [6]. The spatial description is schematized by a representation that constitutes a modeling support for the study of environmental scenes, *i.e.*, 360° scenes perceived by an observer from a fixed vantage viewpoint. The perception of an environment is closely associated to a cognitive organization that reflects different levels of perception [7]. Since humans tend to structure space using distance and bodily directions, an environmental scene is structured by four proximity spaces determined by their distance from the observer, and a directional cone-based partition [8] whose number can vary from two (front/back or right/left) to four (front, back, right and left).

Let us consider an example of scene description given by an observer to a distant addressee who is asked to consider the described environment: " I am on a footpath that runs along a castle and a pond. In front of me, there is a little valley with the castle on the right of it and at the horizon, I can distinguish a mountain range. On my right is the farm of the castle. Behind me, there is the pond with a meadow behind, and a forest far away ". In order to promote communication and cooperation, the observer contribution should be informative enough [9]. We assume that the resulting scene description can result from a preliminary complex dialogue between the observer and the addressee for the settlement of inconsistencies and vagueness problems. In order to keep the rele-

vant information, the description is first parsed and semi-automatically filtered by the Tinky parser [10]. Co-references are identified and resolved, and the description is modeled as a set of triplets u_i such as $u_i = [e_j, r_k, e_l]$ with $e_j, e_l \in \mathcal{E}$ the set of entities of an environmental scene, and $r_k \in \mathcal{R}$ the set of spatial relations. Distance and directional relations are interpreted with the use of an application ontology, and the identified entities are associated to one-to-many proximity spaces and directional cones. No matter the nature, prominence or familiarity of the considered objects, the spatial relations are interpreted in order to favor their relative ordering.

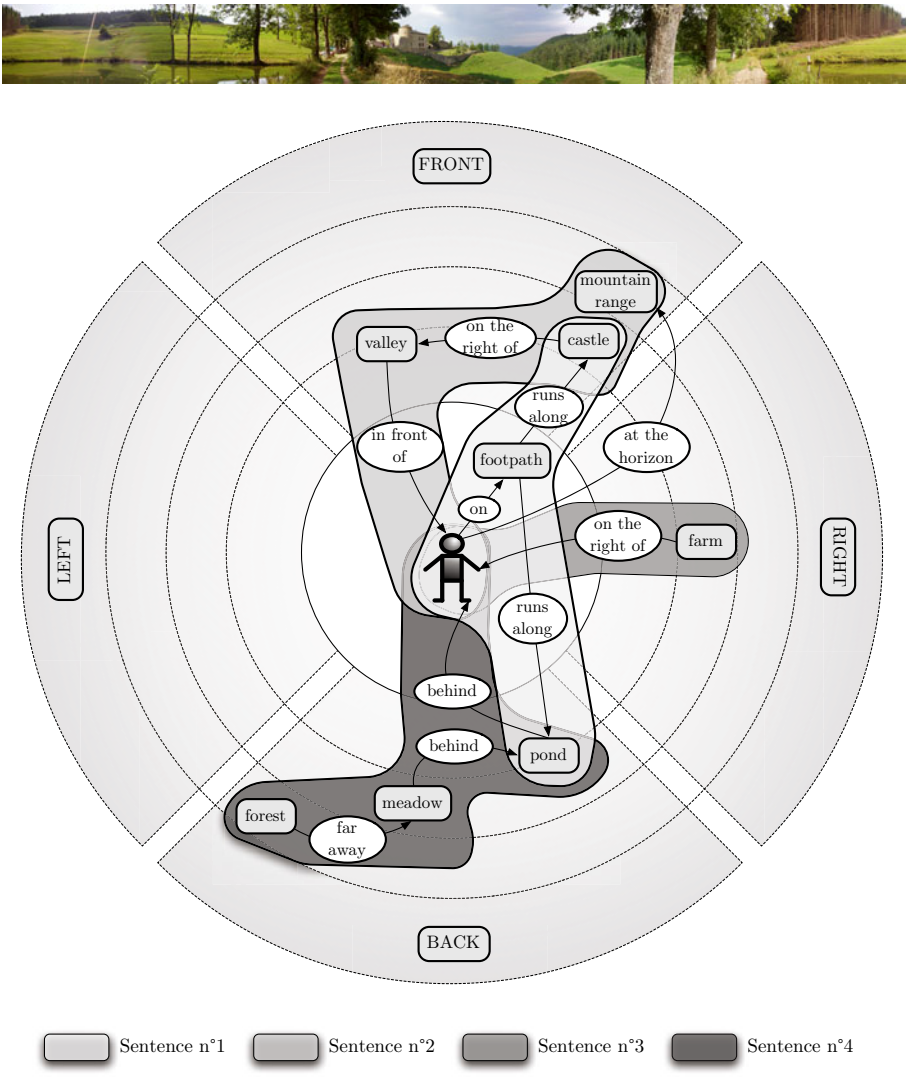


Fig. 1. Conceptual map of a scene [6]

The resulting conceptual map of Figure 1 illustrates the spatial structure, diversity and relative ordering of the entities. Such a model qualifies and characterizes natural landscapes, and provides a framework for the analysis of the properties of the verbal descriptions made by different observers, and cross-comparisons of different landscape descriptions. However, the salient entities of the scene are not always clearly revealed. This has motivated the integration of salience scores, supported by a mutual reinforcement algorithm. It reflects the particularities of the entities that emerge from a scene description such as their linguistic properties, *i.e.*, the richness of information associated to each term, and their structural characteristics, *i.e.*, the degree of spatial isolation [11].

3 Geopositioning Approach

Recent years have witnessed significant geopositioning developments, particularly when locations are not available using global positioning systems. This has been applied to search-and-rescue operations where the location of a human (usually lost) should be retrieved. The ability of lost persons to precisely describe their perceived environment is essential to a successful identification of their location. The way people and particularly children behave and model their environment has been studied by cognitive studies, while cognitive distortions and reasons for retrieval failures have been qualified [12]. Preliminary experiments have explored the potential of GIS for the management of search teams [13], or the behavior of a lost person regarding her initial displacement plans, goals and own abilities [14]. However, GIS have not been used to the best of our knowledge for geopositioning an observer from a qualitative description and interpretation of his environment.

The description of the surroundings of the lost person is used as the only input of our approach. No assertion is made on their background, will, route, the reason why they planned the excursion, etc. The methodology used for the search of the observer results from the analysis of the verbal description. This is based on the interpretation of the entities and landmarks identified in the description, the spatial structure emerging from the proximity spaces that illustrate a relative ordering between the entities, and direction relations between the entities.

The geopositioning approach searches for the possible locations of the observer, and is applied as follows:

1. For each entity identified by the observer, computation of the viewsheds *i.e.*, places for which each quoted entity can be perceived by the observer.
2. Interpretation of the direction relations derived from the directional cones and identification of a new set of candidate solutions for the observer location.
3. Interpretation of the distance relations derived from the proximity spaces and identification of a new set of candidate solutions for the observer location.
4. Interpretation of the direction relations given relatively to entities, and identification of a new set of candidate solutions for the observer location.

3.1 Visibility-Based Approach

Viewshed computation and analysis have long been applied in landscape and urban studies [15,16,17]. The main principle of a viewshed analysis is commonly determined by defining one location as the viewing point, and then calculating the line-of-sight to every other point within the region of interest. When the surface rises above the line of sight the target is out of sight, otherwise it is considered as in-sight [18,19]. The range of application is relatively large, from architectural studies where visibility represents a qualitative parameter for a site selection to minimize or maximize [20], to the distribution of forest-fire observation towers in natural environments [21].

The modeling approach should identify the possible locations of the entities quoted in the description. Let us assume that the described entities are directly visible from the fixed viewpoint of the observer. The visibility-based approach mainly focusses on the area from which the location of the entities can be viewed, as opposed to the visible area that is not equivalent because the height of the object at the viewing point may be different from the height of the viewed object [16]. Our objective is not to precisely locate the observer but rather the area the observer is supposed to be located at. Without loss of generality, we consider as equivalent the regions seen from a given entity and the ones visible from the observer.

Let us introduce the formal representation of the visibility-based approach. Let $\mathcal{E}$ be the set of entities e_i identified in a verbal description D, $\mathcal{S}$ denotes the ordered set of salient entities $e_i \in \mathcal{E}$, *i.e.*, $\mathcal{S} = [e_1, e_2, \ldots, e_n]$ and n denotes the cardinality of set $\mathcal{S}$. Let $\mathcal{B}$ be the set of objects of a GIS database considered as the repository of the region of interest. These objects and the relations materialized in this GIS database are organized into a lattice of classes and sub-classes, that result from a classification provided by the French Institut Géographique National (Fig. 2).

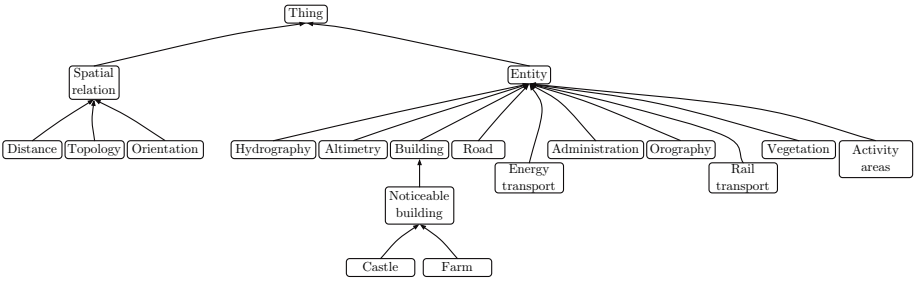


Fig. 2. Top-level concepts of the application taxonomy

Let $\mathcal{C}$ be the set of classes of the GIS database. The function f_{class} that associates the class $c \in \mathcal{C}$ to an object $b_j \in \mathcal{B}$ is given by

$$\begin{aligned} f_{class} : \mathcal{B} &\rightarrow \mathcal{C} \\ b_j &\mapsto c. \end{aligned} \tag{1}$$

Similarly, the function g_{class} that associates the class $c \in \mathcal{C}$ to an entity $e_i \in \mathcal{S}$ is given by

$$\begin{aligned} g_{class} : \mathcal{S} &\rightarrow \mathcal{C} \\ e_i &\mapsto c. \end{aligned} \tag{2}$$

Let $e_i \in \mathcal{S}$ a salient entity of the verbal description. Since entity e_i is visible by the observer, the visibility-based approach should select all objects b_j of the database that correspond to the class of objects identified in the description as all could potentially be perceived by the observer. Let $\mathcal{R}$ denote the set of m objects $b_j \in \mathcal{B}$ whose class fits that of entity e_i, *i.e.*, $g_{class}(b_j) = f_{class}(e_i)$, and $i \in [0, \ldots, m]$, $j \in [0, \ldots, n]$.

Let v be the function dedicated to the visibility computation. Let t be a digital terrain model, *i.e.* a triangulated irregular network that describes the topography of the region of interest. Given an object b_j located on terrain t, the viewshed of b_j is the set of points p of t from which b_j is visible. We consider that two points are defined as being visible to each other if a straight line can be drawn between the points without intersecting any part of the terrain surface between them, *i.e.*, $v(b_j, t) = \{\ \mathrm{p} \in \mathrm{t}\ /\ [b_j, \mathrm{p}] \cap \mathrm{t} = \emptyset\ \}$. Let $h_{vis}(e_i)$ be the function that corresponds to the set of areas from which objects b_j, $j \in [1, \ldots, m]$ of a similar class than entity e_i are visible. Then, $h_{vis}(e_i) = \{v(b_1, t), \ldots, v(b_m, t)\}$.

Let S_{vis} be a function that computes the possible locations of the observer that result from the visibility-based approach. S_{vis} is defined by the intersection of the different viewsheds associated to each salient entity, *i.e.*, $\mathbf{S}_{vis} = \bigcap\{h_{vis}(e_i)\}$, where $i \in [1, \ldots, n]$. It is worth noting that the salience-based approach enables to consider only the most salient entities rather than considering all of them. Afterwards, we shall however take into account all entities.

Let us consider the example of description " I am on a *footpath* that runs along a *castle* and a *pond*. In front of me, there is a little *valley* with the *castle* on the right of it " and the resulting ordered set of salient entities $\mathcal{S}$={"castle", "footpath", "pond", "valley"}. Firstly, the m objects b_j of the database whose class is "castle" are selected and each viewshed $v(b_j, t)$ is computed. If object "castle" is the one considered, a solution for the possible location region of the observer is given by the set of viewsheds of b_j's, *i.e.* $h_{vis}(castle) = \{v(b_1, t), \ldots, v(b_m, t)\}$. The same method is applied to objects "footpath", "pond" and "valley", and the visibility-based solution is provided by the intersection of the viewsheds associated to each salient entity, *i.e.*, $\mathbf{S}_{vis} = \{h_{vis}(\text{“}castle''\text{)} \cap h_{vis}(\text{“}footpath''\text{)} \cap h_{vis}(\text{“}pond''\text{)} \cap h_{vis}(\text{“}valley''\text{)}\}$. The candidate objects are those that can be perceived directly from the area given by the visibility-based solution, *i.e.*, $\{b_j, c_j, d_j, e_j$ such as $b_j \in$ “$castle''$, $c_j \in$ “$footpath''$, $d_j \in$ “$pond''$, $e_j \in$ “$valley''$ *and* $\bigcap(h_{vis}(b_j), h_{vis}(c_j), h_{vis}(d_j), h_{vis}(e_j)) \neq \emptyset\}$.

The visibility-based approach identifies the possible location areas where the observer can be located. Figure 3 summarizes the principle of this approach. This first step also identifies a set of physical objects of the environment that could potentially be observed from the location regions $\mathbf{S}_{vis}$. This preliminary filtering of the solution area will be refined in the following sub-sections by the

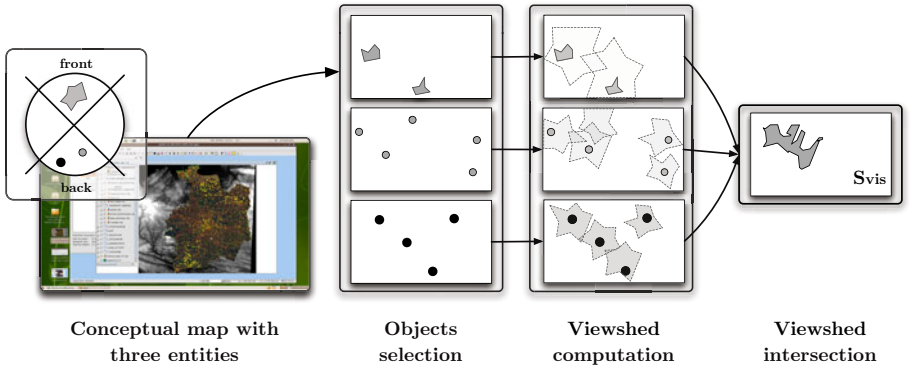


Fig. 3. Visibility principle

interpretation of spatial relations between these objects and illustrated by the proximity spaces and directional cones.

3.2 Spatial Relations as Spatial Constraints

Spatial relations are interpreted as spatial constraints that refine the possible locations of the observer. These spatial constraints are derived from the interpretation of linguistic expressions, and supported by the use of directional cones and proximity spaces of the conceptual map. This can be considered as a specific application of declarative modeling that is commonly applied to the automatic generation of an environment that corresponds to some linguistic descriptions. In particular, text-to-scene modeling has been used in many applications such as architectural design [22], and for the generation of virtual urban landscapes and animated scenes from road accident reports [23]. These modeling approaches are based on the interpretation of spatial constraints and semantic knowledge [24]. This is equivalent to a constraint satisfaction problem applied to the linguistic relations identified in an environment description [25]. A constraint-solver algorithm analyses the coherence or incoherence of the linguistic description in order to derive a possible representation of the scene.

The interpretation of the spatial relations quoted in the description is supported by the directional cones and proximity spaces that structure the conceptual map (Fig. 1). The principle consists in finding the limits of the location of the observer for which:

- The entity locations in the directional cones fulfill the linguistic description properties (Cases 1 and 2).
- Distance relations given relatively to the observer are geometrically interpreted and supported by the use of proximity spaces (Case 3).
- Relative direction relations between two entities are geometrically interpreted (Case 4).

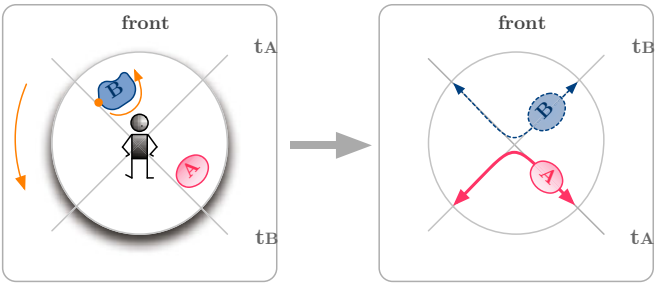


Fig. 4. Orientation constraint principle

Case 1. Entities in opposite directional cones. Let us consider two entities located into two opposite directional cones (front/back or right/left), *i.e.*, entities related to the observer by two opposite direction relations. The algorithm identifies the possible locations of the observer by computing the limits where object B is in front of the observer and object A is behind him, with a space segmented by two straight lines t_A and t_B that define the four directional cones, *i.e.* front, back and right, left. Objects A and B of the database are fixed, and the principle illustrated by Figure 4 consists in identifying the limits of the solution that satisfies the directional constraints by moving the directional cones along the whole boundary of object B. This generates a relative displacement of object A while retaining its location in the back cone, and object B in the front cone. In order to get an exhaustive solution without losing possible locations due to approximation errors, we consider that an object that intersects the boundary of a directional cone is in that directional cone. In such a case, t_A (resp. t_B) is tangent to object A (resp. B).

Let us consider two surface objects A and B (Fig.5), and P_B (resp. P_A) the tangent point to B and tangent line t_B (resp. t_A). In order to identify the observer location for which object B is in front of him, and object A behind, space is discretized by uniformly moving point P_B (resp. P_A) along the boundary of object B (resp. A). For each location of point P_B (resp. P_A),

1. Construction of tangent t_B (resp. t_A).
2. Construction of the exterior tangent t_A to object A (resp. t_B to object B) that is also perpendicular to t_B (resp. t_A) [1].

The intersection O_i of t_A with t_B gives a boundary point of the solution, *i.e.*, a boundary point for the possible location of the observer. Consequently, the region $\mathcal{A}(object_A, object_B)$ in which the observer should be located is bounded by the convex hull materialized by points $O_1, \ldots, O_m$ with $m \in \mathbb{N} \setminus \{0\}$. The algorithm is similar when objects A and/or B are modeled as polylines or points. The difference with polylines is given by the intersection between t_B and B (resp. t_A and A) that can then be a segment rather than a point.

[1] The interior tangent is not necessary since the resulting points are included in the solution.

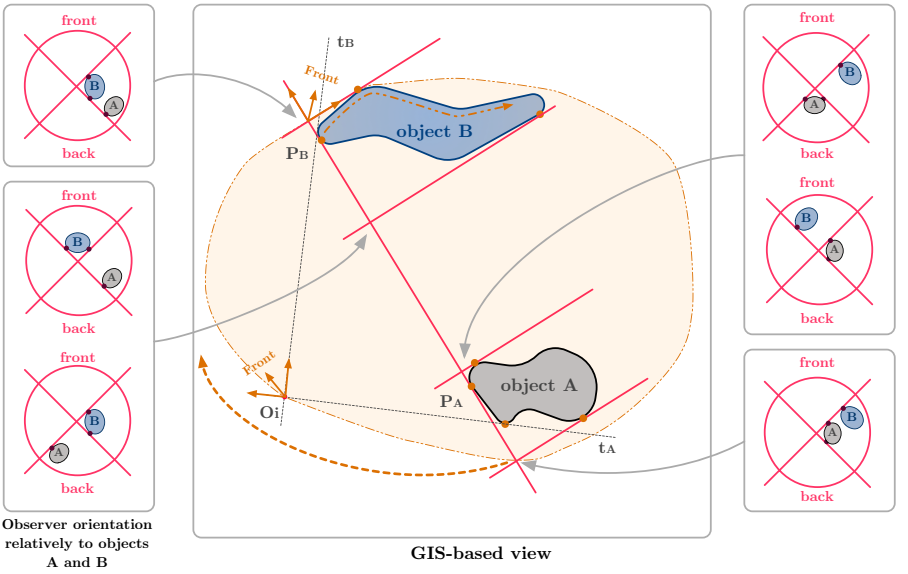


Fig. 5. Orientation constraint

Figure 5 illustrates the possible observer locations with respect to two objects A and B, when considering their relative locations to the observer. The solutions presented to the left and right of the figure show the range of the observer positions that fulfill the direction constraints. This algorithm is applied to all object pairs of the geographical database whose classes correspond to those of the entities located in two opposite directional cones. Let us consider the following example " the castle is in front of me, and the valley behind ", the previous algorithm is then applied to all object pairs ("castle", "valley") of the geographical database that also belong to the previous solution S_{vis}. Let $\mathbb{E}_{C1}$ be the set of entities of the verbal description that are in a directional cone (*e.g.* $\mathbb{E}_{front}$), and $\mathbb{E}_{C2}$ the set of entities of the verbal description that are in the opposite cone (*e.g.* $\mathbb{E}_{back}$), $\mathbb{E}_{C1/2}$ the set of combinations of entity pairs (e_i, e_j) of the verbal description with $e_i \in C1$ and $e_j \in C2$, $\mathbb{O}_{C1}$, $\mathbb{O}_{C2}$, $\mathbb{O}_{C1/2}$ the corresponding set of objects and combinations of objects (o_m, o_n) of the GIS, and $\mathcal{A}(o_m, o_n)$ the solution region with $o_m \in \mathbb{O}_{C1}$ and $o_n \in \mathbb{O}_{C2}$. The solution region $S_{opposite\_rel}$ for which some objects are in a cone (*e.g.* front) and others in the opposite one (*e.g.* back) is the union of regions $\mathcal{A}(o_m, o_n)$. When space is partitioned with four directional cones, a possible solution $\mathbf{S}_{dir}$ is given by the intersection of the two sub-solutions regions that correspond to the solutions front/back and right/left.

Let N_B and M_B denote the intersections of the interior tangents t_A and t_B with object B. When space is partitioned by two subspaces, *i.e.*, front/back or right/left, the solution is given by uniformly moving point P_B along the boundary of object B from points N_B to M_B (Fig 6). For each P_B, a solution $\mathcal{S}^i_{P_B}$ given by the half space behind object B fulfills the constraint for which B is in front of the observer (*i.e.*, the observer is behind B). A similar method is applied

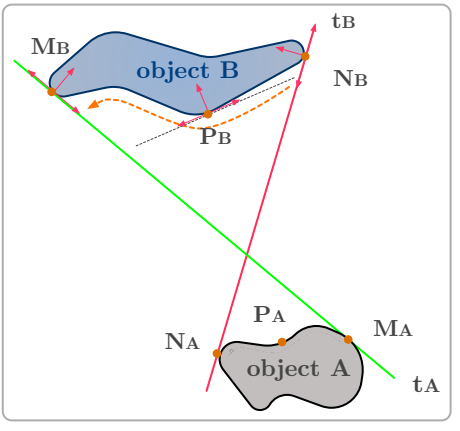


Fig. 6. Orientation constraint (two cones)

to object A. For each P_A, a solution $\mathcal{S}^i_{P_A}$ given by the half space is front of object A. $\mathbf{S}_{dir}$ corresponds to the intersection of the two sub-constraints, *i.e.*, $\mathbf{S}_{dir} = \cap(\mathcal{S}^i_{P_B}, \mathcal{S}^i_{P_B})$.

Case 2. Entities in an identical cone. Let us consider a second case where two entities are located in a same directional cone, *i.e.*, entities related to the observer in a similar way. An equivalent algorithm is applied with the difference that it identifies the possible locations of the observer by computing the limits for which objects A and B are in a same cone. The solution region $\mathbf{S}_{dir\_cone}$ is the complement of solution given by Case 1.

When space is partitioned by two directional cones (front/back or right/left), the solution is constructed by uniformly moving point P_B (resp. P_A) along the boundary of object B (resp. A) from points N_B to M_B (resp. N_A to M_A) that constitutes the intersection of the exterior tangents t_A and t_B to object B (resp. A). The solution is given by the exterior region bounded by the convex hull between A and B.

Case 3. Distance relations. Distance relations, whether used in an egocentric or allocentric manner, are also integrated in the geopositioning process. Their interpretation is given by the use of the proximity spaces that structure the conceptual map. When some entities are located in an identical directional cone, they can be located in different or similar proximity spaces with respect to the location of the observer. Figure 7 illustrates an example of distance constraint satisfaction where object A is in front of the observer, and B is behind A. The space $\mathcal{S}_{distance}$ that satisfies the two constraints also refers to the possible locations of the observer. The principle consists in finding the limits of the observer's location with respect to the following constraints:

- Entities A and B are in an identical directional cone.
- Their distances relatively to the observer correspond to those supported by the conceptual map.

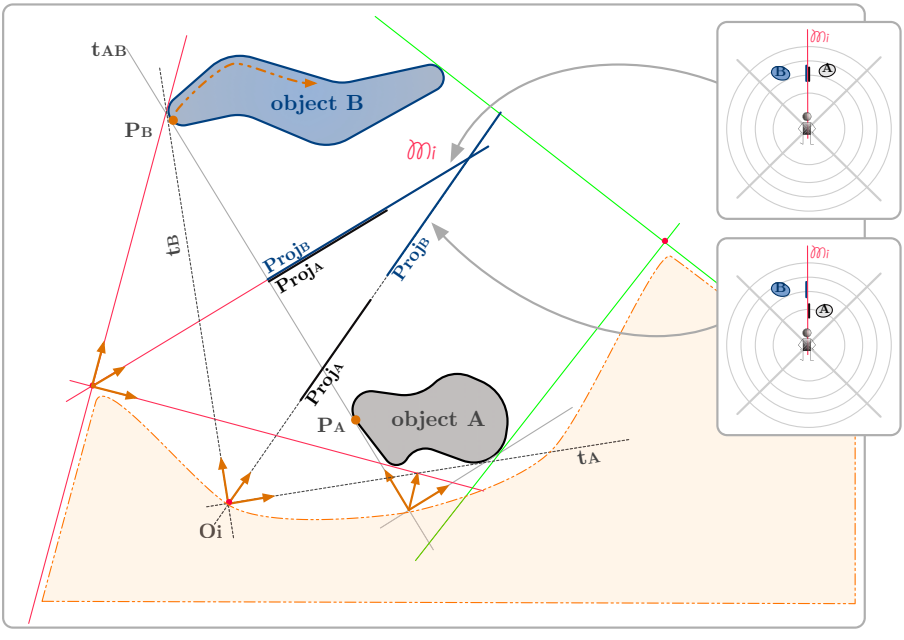


Fig. 7. Distance constraint

When searching for a candidate location of the observer relatively to objects A and B as illustrated in the previous example, the search method is as follows. Space is uniformly discretized by uniformly moving point P_B (resp. P_A) along the boundary of object B (resp. A). For each location of point P_B (resp. P_A), tangents t_B and t_A are constructed. The relative ordering of the different entities composing the environmental scene is defined by projecting entities A and B on the median line that bisects angle $\widehat{t_A, t_B}$ (Fig. 7). The interval endpoints are compared one to the other, as applied elsewhere in unidimensional spaces [26]. We consider that B is behind A when their endpoint beginnings (relatively to the observer) coincide. This search algorithm is similarly applied when the observed entities are represented as polylines. The difference is given by the fact that the intersection between t_B and B (resp. t_A and A) can be a segment line rather than just a point. However, it is worth noting that a partition of space based on two directional cones does not enable the identification of a candidate solution.

Case 4. Ternary direction relations. Let us consider the case of a ternary relation using an observer-centered frame of reference [27], *i.e.*, *"on the left of"* or *"on the right of"* between two distinct entities of an identical directional cone. If the observer identifies object A as being on the left of B, it means that he/she is in a half-space defined relatively to the location of A and B. Let t_{AB} and t'_{AB} the exterior tangents of objects A and B, and $(M, \overrightarrow{i}, \overrightarrow{j})$ and $(N, \overrightarrow{k}, \overrightarrow{l})$ the basis as illustrated by Figure 8. On the one hand, if "A is on the left of B", a

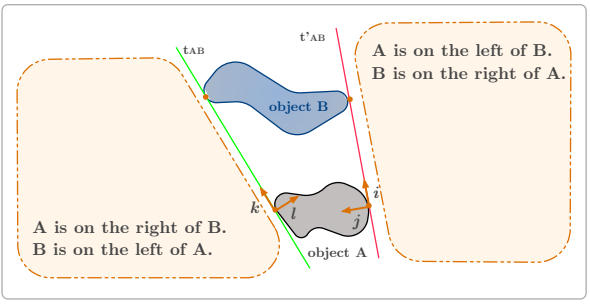


Fig. 8. Relative direction constraint

solution is provided by the half-space such as $S_{ternary} = \{\overrightarrow{j} < 0\}$ .On the other hand, if "A is on the right of B", a solution is provided by the half-space such as $S_{ternary} = \{\overrightarrow{l} < 0\}$. The algorithm is similar when entities A and/or B are modeled as polylines or points.

Integration of the successive results. The successive constraints can be summarized as follows:

- $\mathcal{S}_{dir}$ is the space for which opposite direction constraints given relatively to the observer between two entities are satisfied.
- $\mathcal{S}_{dir\_cone}$ is the space for which location constraints between two entities of a similar directional cone are satisfied.
- $\mathcal{S}_{distance}$ is the space for which distance constraints between two entities of a same cone are satisfied.
- $\mathcal{S}_{ternary}$ is the space for which direction constraints given relatively to two entities are satisfied.

Overall, the final solution, *i.e.* the areas that correspond to the possible observer's locations are given by the intersection of the solutions provided by each of these constraints. Since the geopositioning algorithm successively applies these complementary constraints, it significantly reduces the size of the solution space. Let us consider the spatial configuration given by the verbal description " I am in front of a meadow and the castle is behind me. The drawbridge is behind the castle " and illustrated by the conceptual map of figure 3. The application of the parser leads to the identification of three triplets, *e.g.*, [meadow, in-front-of, observer], [castle, behind, observer] and [drawbridge, behind, castle]. The visibility algorithm identifies a first solution region $\mathbf{S}_{vis}$ and the corresponding object candidates. Four spatial constraints emerge from the previous example and are applied to each candidate object:

- the first case where *entities are in opposite directional cones* is applied both on pairs ("meadow", "castle") and ("meadow", "drawbridge").

– the second case where *entities are in an identical cone* is applied on the pair ("castle", "drawbridge").
– the third case that characterizes a distance relations between entities of a same cone is applied on the pair ("castle", "drawbridge").

Consequently, four possibles solution spaces $\mathcal{S}_{dir}\_1$, $\mathcal{S}_{dir}\_2$, $\mathcal{S}_{dir\_cone}$ and $\mathcal{S}_{distance}$ emerge. The final solution is given by their intersection including the previous solution $\mathcal{S}_{vis}$ (Fig. 9).

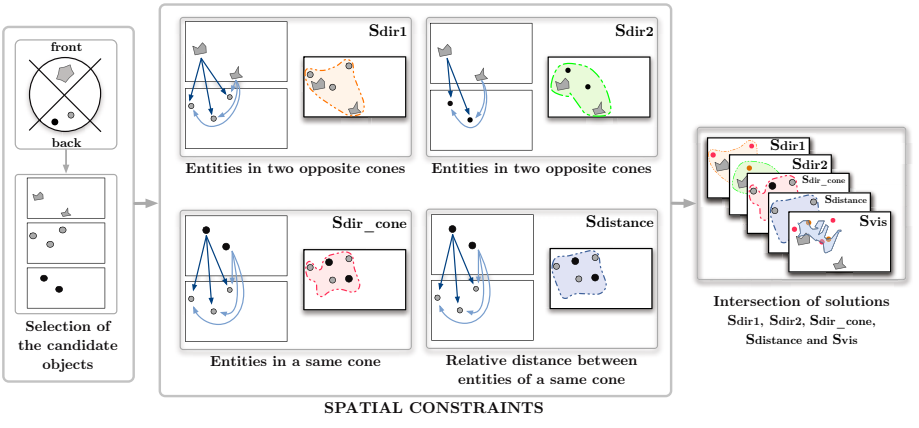


Fig. 9. Constraint-based approach

4 Conclusion

Early models of geopositioning processes have been widely influenced by quantitative representations of space. However, these approaches do not completely reflect the way humans perceive and describe their environment since they preferably process qualitative information. This paper introduces a method for geopositioning an observer from the verbal description of his/her surroundings. A constraint-satisfaction algorithm is applied by successively refining the candidate locations of the observer. The first case of the approach considers some visibility constraints on the entities identified in the verbal description with respect to some candidate objects of the geographical database. The second case considers direction and distance relations as spatial constraints, *i.e.* relative directions between entities and the observer, as well as distance relations are interpreted. Overall the geopositioning approach provides a set of possible locations for the observer. The algorithm still deserves integration of additional spatial relations, such as non-visibility constraints that can be derived from entities and landmarks not identified by the observer, but present in the geographical database. The approach is currently being implemented as an extension of the *GvSIG* software.

References

1. Herskovits, A.: Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English. Cambridge University Press, Cambridge (1986)
2. Tversky, B., Lee, P.U.: How space structures language. In: Freksa, C., Habel, C., Wender, K.F. (eds.) Spatial Cognition 1998. LNCS (LNAI), vol. 1404, pp. 157–175. Springer, Heidelberg (1998)
3. Kuipers, B.: Modeling spatial knowledge. Cognitive Science 2(2), 129–153 (1978)
4. Freksa, C.: Using Orientation Information for Qualitative Spatial Reasoning. In: Frank, A.U., Formentini, U., Campari, I. (eds.) GIS 1992. LNCS, vol. 639, pp. 162–178. Springer, Heidelberg (1992)
5. Smith, B., Mark, D.: Geographical categories: An ontological investigation. International Journal of Geographical Information Science 15(7), 591–612 (2001)
6. Le Yaouanc, J.M., Saux, E., Claramunt, C.: A semantic and language-based representation of an environmental scene. GeoInformatica 14(3), 333–352 (2010)
7. Montello, D.: Scale and multiple psychologies of space. In: Campari, I., Frank, A.U. (eds.) COSIT 1993. LNCS, vol. 716, pp. 312–321. Springer, Heidelberg (1993)
8. Peuquet, D.J., Ci-Xiang, Z.: An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. Pattern Recognition 20(1), 65–74 (1987)
9. Grice, H.: Logic and conversation. Syntax and Semantics 3(S 41), 58 (1975)
10. Ligozat, G., Nowak, J., Schmitt, D.: From language to pictorial representations. In: Poznańskie, W. (ed.) Proc. of the Language and Technology Conference, Poznan, Poland (September 2007)
11. Le Yaouanc, J.M., Saux, E., Claramunt, C.: A salience-based approach for the modeling of landscape descriptions. In: Agrawal, D., Lu, C.T., Wolfson, O. (eds.) Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 396–399. ACM Press, New York (2009)
12. Cornell, E.H., Hill, K.A.: The problem of lost children. In: Children and their Environments: Learning, Using, and Designing Spaces, pp. 26–41. Cambridge University Press, Cambridge (2005)
13. Heth, C.D., Cornell, E.H.: A Geographic Information System for Managing Search for Lost Persons. In: Applied Spatial Cognition: From Research to Cognitive Technology. Lawrence Erlbaum Associates, Mahwah (2006)
14. Ferguson, D.: GIS for wilderness search and recue. In: Proceedings of ESRI Federal User Conference (2008)
15. De Floriani, L., Marzano, P., Puppo, E.: Line-of-sight communication on terrain models. International Journal of Geographic Information Systems 8(4), 329–342 (1994)
16. Fisher, P.F.: Extending the applicability of viewsheds in landscape planning. Photogrammetric Engineering & Remote Sensing 62(11), 1297–1302 (1996)
17. Fogliaroni, P., Wallgrün, J.O., Clementini, E., Tarquini, F., Wolter, D.: A qualitative approach to localization and navigation based on visibility information. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 312–329. Springer, Heidelberg (2009)
18. De Floriani, L., Magillo, P.: Algorithms for visibility computation on terrains: a survey. Environment and Planning B (Planning and Design) 30(5), 709–728 (2003)
19. Lee, J.: Analyses of visibility sites on topographic surfaces. International Journal of Geographic Information Systems 5, 413–429 (1991)

20. Smardon, R.C., Palmer, J.F.: Foundations for Visual Project Analysis. Wiley and Sons, Inc., Chichester (1986)
21. Travis, M.R., Elsner, G.H., Iverson, W.D., Johnson, C.G.: VIEWIT: computation of seen areas, slope, and aspect for land-use planning. Technical Report GTR-PSW-011, Berkeley, CA: Pacific Southwest Research Station, Forest Service, U.S. Department of Agriculture (1975)
22. Larive, M., Dupuy, Y., Gaildrat, V.: Automatic generation of urban zones. In: Kunii, T.L., Skala, V. (eds.) Proceedings of Computer Graphics, Visualization and Computer Vision (2005)
23. AAkerberg, O., Svensson, H., Schulz, B., Nugues, P.: CarSim: an automatic 3D text-to-scene conversion system applied to road accident reports. In: Copestake, A., Hajic, J. (eds.) Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, vol. 2, pp. 191–194. ACM Press, New York (2003)
24. Le Roux, O., Gaildrat, V., Caubet, R.: Using Constraint Satisfaction Techniques in Declarative Modelling. In: Geometric Modeling: Techniques, Applications, Systems and Tools, pp. 1–20. Springer, Heidelberg (2004)
25. Desmontils, E.: Expressing constraint satisfaction problems in declarative modeling using natural language and fuzzy sets. Computers and Graphics 4(24), 555–568 (2000)
26. Mukerjee, A.: A representation for modeling functional knowledge in geometric structures. In: Ramani, S., Anjaneyulu, K.S.R., Chandrasekar, R. (eds.) KBCS 1989. LNCS, vol. 444, pp. 192–202. Springer, Heidelberg (1990)
27. Levinson, S.: Frames of reference and Molyneux's question: cross-linguistic evidence. Language and Space, 109–169 (1996)

Area-Preserving Subdivision Schematization*

Wouter Meulemans, André van Renssen, and Bettina Speckmann

Dep. of Mathematics and Computer Science, TU Eindhoven, The Netherlands
{w.meulemans,a.m.v.renssen}@student.tue.nl
speckman@win.tue.nl

Abstract. We describe an area-preserving subdivision schematization algorithm: the area of each region in the input equals the area of the corresponding region in the output. Our schematization is axis-aligned, the final output is a rectilinear subdivision. We first describe how to convert a given subdivision into an area-equivalent rectilinear subdivision. Then we define two area-preserving contraction operations and prove that at least one of these operations can always be applied to any given simple rectilinear polygon. We extend this approach to subdivisions and showcase experimental results. Finally, we give examples for standard distance metrics (symmetric difference, Hausdorff- and Fréchet-distance) that show that better schematizations might result in worse shapes.

Keywords: Schematization, polygonal subdivisions.

1 Introduction

A schematic map displays a set of nodes and their connections—for example, highway, train, or metro networks—in a highly simplified form to communicate the connectivity information as effective as possible. Connections are usually drawn as polygonal paths using few links and few orientations; the orientations are restricted to be axis-parallel or to adhere to the four main orientations.

Although most previous efforts are concentrated on the schematization of networks, it is of course also possible, and often desirable, to schematize the boundaries of regions or even complete subdivisions. This is particularly useful in conjunction with schematic networks. Consider, for example, a schematized railway network which is displayed on top of a geographic base map. A detailed depiction of the region's boundary distracts from the schematic map, whereas a schematized version supports the schematic map (see Fig. 1). Schematized regions are also used when depicting fare zone boundaries. Generally, whenever exact boundaries

Fig. 1. Danish rail network [http://www.eurail.com/]

* B. Speckmann is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 160–174, 2010.

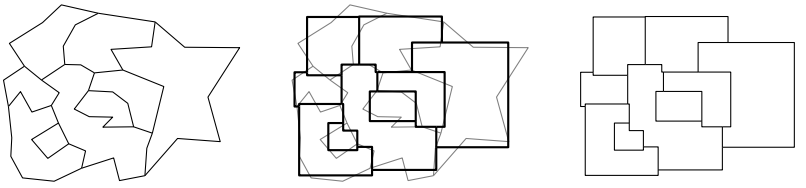


Fig. 2. A subdivision and an area-preserving schematization

are not needed it is preferable to replace them by schematic ones, to reduce visual clutter and indicate that the purpose of the map in question is not a (purely) geographic one.

A schematized subdivision of high visual quality satisfies at least the following criteria. Regions are approximated using few links and few orientations. There are no self-intersections and the region adjacencies of the input map are maintained. Finally, the output visually resembles the input, that is, region shapes and sizes are preserved as well as possible. It is comparatively easy to avoid self-intersections and to ensure proper adjacencies. However, it is less clear how to create regions of the "best" shape.

Results. We focus on area-preserving schematization, that is, the area of each region in the input subdivision equals the area of the corresponding region in the output. In particular, we present a schematization algorithm which is based on two simple area-preserving contraction operations. Our schematization is axis-aligned, the final output is a rectilinear subdivision (see Fig. 2). Our contraction operations are defined for rectilinear polygons. Hence our first step is to convert a given input subdivision into an area-equivalent rectilinear subdivision, see Section 2 for details. Experiments show that our rectilinearization approach increases the number of edges only by a small constant factor.

In Section 3 we discuss our contraction operations in detail. We prove that any given rectilinear polygon with 6 or more edges can be simplified in an area-preserving manner. That is, at least one of our operations can always be executed. We then extend our approach to subdivisions and show how our operations can be adapted to vertices of degree 3. For simple polygons we can guarantee any desired output complexity; this is not the case for subdivisions. However, our experiments show that also subdivisions can be schematized using few edges.

For a given input subdivision there are clearly many area-preserving schematizations of equal complexity (number of edges). It would seem natural to choose the best of those with respect to any standard distance metric, such as the symmetric difference, the Hausdorff-distance, or the Fréchet-distance. However, in Section 4 we show examples for each of these distance functions where better approximations result in worse shapes. Hence our algorithm does not try to minimize either of these distance functions but instead greedily contracts those parts of the subdivision where the contraction results in the smallest symmetric difference. Finally, in Section 5 we showcase some results of our algorithm.

Related Work. There is an ample body of work on map schematization and metro map construction. For example, Cabello *et al.* [1] give an algorithm that schematizes a given network using two or three links per path, if that is possible. Nöllenburg and Wolff [2] use a method based on mixed-integer programming to generate metro maps using one edge per path. Swan *et al.* [3] give an extensive overview of existing schematization algorithms and study their applicability to automated schematic map construction for web services. Algorithms for map schematization can be used to schematize subdivisions. However, they usually do no take criteria such as shape and size preservation into account.

Cartographic generalization is a very active research field, with a multitude of new results appearing each year. Of particular relevance to this paper is the generalization of urban data, specifically building generalization [4,5,6,7]. Building generalization typically involves several consecutive actions or operators. Among those, building wall squaring [8,9,10] and outline simplification [9,11,12] are most closely related to our work. Algorithms for these generalization tasks can also be used for subdivision schematization and vice versa; in Section 5 we show a few examples of building outlines generalized with our algorithm.

Line simplification has been a prominent topic in the GIS literature for many years and various quality criteria have been proposed. One of the possible criteria is areal displacement; see the work by Bose *et al.* [13] for an area-preserving approach. When simplifying subdivisions it is generally not advisable to simplify each chain of the subdivision in isolation. There are some approaches, developed in computational geometry, that preserve the topology of the input subdivision. De Berg *et al.* [14,15] describe a method that simplifies a polygonal subdivision without introducing intersections or passing over special input points. Estkowski and Mitchell [16] give a heuristic for simplifying parallel lines, such as elevation contours. Van de Kraats *et al.* [17] discuss the special case where the subdivision to be simplified is a printed circuit board. Unfortunately many subdivision simplification problems that minimize the output complexity are NP-complete [18].

2 Rectilinearization

Here we describe how to turn a simple subdivision S into a simple *area-equivalent* rectilinear subdivision R. That is, R is a simple rectilinear subdivision, the regions of R correspond one-to-one to the regions of S, the adjacencies between regions are maintained, and the area of each region in R equals the area of its corresponding region in S. We assume that the input subdivision S has vertex degree at most three. The *complexity* (number of edges) of the output subdivision R depends on the minimal distance δ between a vertex and an edge of S.

The quality of the schematization improves if R does not contain "long" edges. We use a small constant fraction $\alpha \approx 0.002$ of the diameter of S as an upper bound for the edge length and split all edges of S which are longer.

We associate four axis-aligned *quadrants* with each vertex v of S. We call a vertex v of an edge (u, v) *sharp* if

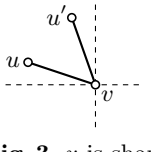


Fig. 3. v is sharp

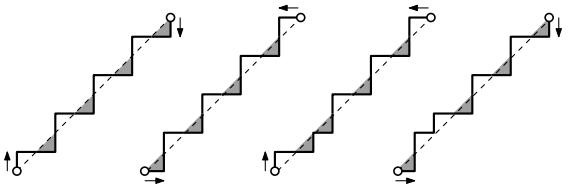


Fig. 4. An edge is rectilinearized using 8 steps

there is another edge (u', v) such that u and u' lie in the same quadrant of v (see Fig. 3). Let $e = (u, v)$ be an edge of S. We assign an axis-aligned direction (up, down, left, or right) to each pair $\langle e, u\rangle$ and $\langle e, v\rangle$ independently. Since each vertex has degree at most three we can easily find an assignment of directions to its outgoing edges which ensures that (i) no two edges are assigned the same direction, and (ii) the total angular deviation is minimized. We assume for ease of explanation that not all three outgoing edges of a vertex lie in the same quadrant. Our approach can be adapted to also deal with this case (and with vertices of degree four), but at a substantial increase of cases to be considered.

We now show how to rectilinearize each edge within its axis-aligned bounding box in an area-preserving manner while avoiding intersections. We first consider an edge e without a sharp vertex. Let $d(e)$ denote the minimal distance between e and any edge that intersects the axis-aligned bounding box of e. We ensure that the maximal distance between e and its rectilinearization is at most $d(e)/2$, which implies that we do not introduce intersections. We rectilinearize e by making $s(e)$ many *steps*: a step starts on e, then goes horizontally (or vertically) away from e and returns vertically (or horizontally) to e. An area-preserving approximation must take as many vertical as horizontal steps, hence $s(e)$ needs to be even. A step can cover at most a distance $d(e)$ along e, otherwise the distance between e and its rectilinearization exceeds $d(e)/2$. The number of steps hence equals at least the length of e divided by $d(e)$ and rounded up to the next even number. We alternate steps depending on the directions assigned to e at its two vertices, see Fig. 4 for some examples.

Next we consider edges that have a sharp vertex. If an edge e has two sharp vertices, we split it. Let e now be an edge with a sharp vertex v and let e' be the second edge that has v as a vertex and lies in the same quadrant of v. In principle we treat e as before, with two exceptions. The first difference is the computation of $d(e)$. For this, we ignore the first quarter of e starting at v, that is, we compute $d(e)$ as the distance to a shorter edge $\hat{e}$ that coincides with e but

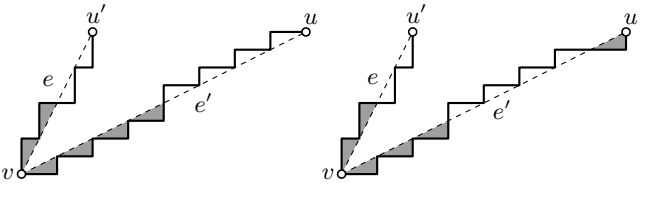


Fig. 5. Rectilinearizing edges with sharp vertices

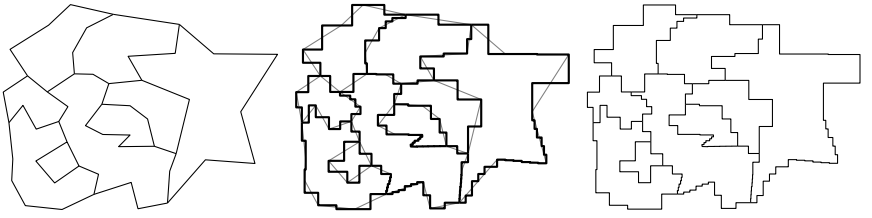


Fig. 6. A subdivision and an area-preserving rectilinearization (with $\alpha = 0.1$)

is missing the first quarter. Secondly, when rectilinearizing e we use "evasive" behavior along the first quarter: all steps lie on the "other" side (see Fig. 5).

The procedure as outlined above turns a simple subdivision S into a simple area-equivalent rectilinear subdivision R in $O(n^2 + m)$ time, where n is the complexity of S and m is the complexity of R. See Fig. 6 for an example. In our experiments the number of edges of the rectilinearized subdivision was always within a constant factor (usually around three, never more than eight) of the number of edges of the input subdivision.

3 Schematization

In this section we describe how to simplify a given rectilinear polygon or subdivision with the help of two area-preserving contraction operations. In Subsection 3.1 we first introduce our operations and then prove that at least one of them can be applied to any simple rectilinear polygon with at least six vertices. In Subsection 3.2 we extend our approach to subdivisions and sketch how to adapt our operations to vertices of degree three.

3.1 Simple Polygons

Assume that we are given a simple rectilinear polygon R. Our area-preserving contraction operations work with configurations of three consecutive edges along the boundary of R. We distinguish *S-configurations* (a left turn followed by a right turn or vice versa) and *C-configurations* (two left turns or two right turns).

The first operation is the *S-contraction*: we replace an S-configuration with the weighted average of its two outer edges and connect this new edge to the previous neighbors of the S-configuration (see Fig. 7). This operation is clearly area-preserving. An S-contraction reduces the complexity of the polygon by at least two. The *contraction area* of an S-configuration is the symmetric difference between the polygon before and after the corresponding S-contraction (indicated in gray in Fig. 7). An S-configuration is *feasible* if its contraction area is empty. Not

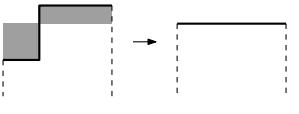


Fig. 7. An S-contraction

surprisingly, an arbitrary rectilinear polygon might not have any feasible S-configurations. In this case we need our second operation, the *C-contraction*, which is based on two complementary C-configurations.

We distinguish two types of C-configurations: *inner C-configurations* (the interior of the polygon lies on the same side of the middle edge as the two outer edges) and *outer C-configurations* (the interior of the polygon lies on the other side of the middle edge as the two outer edges). A C-contraction requires both an inner and an outer C-configuration. We move the middle edges of both C-configurations simultaneously, until the length of one of the outer edges is reduced to zero. A simple calculation shows that we can choose the speed with which to move each edge in such a way that the operation is area-preserving (see Fig. 8). Just as an S-contraction, a C-contraction reduces the complexity of the polygon by at least two. The contraction area of a C-configuration is the rectangle defined by its middle edge and the shorter of its two outer edges. As before, a C-configuration is feasible if its contraction area is empty.

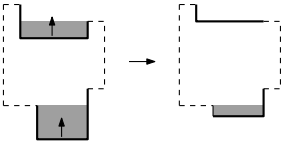


Fig. 8. A C-contraction

We now prove through a sequence of lemmas that every simple rectilinear polygon R with at least six edges has either a feasible S-configuration or two complimentary feasible C-configurations. That is, at least one of our operations can be applied to R as long as R is not a rectangle.

Lemma 1. *Every rectilinear polygon R with at least six edges has a feasible inner C-configuration.*

Proof. Let e_h be the highest horizontal edge of R. The two vertical neighbors of e_h must necessarily be directed down and hence e_h is the middle edge of an inner C-configuration C_h. If C_h is feasible, then we are done. Otherwise let e'_h be the highest horizontal edge in the contraction area of C_h. We distinguish two cases: (i) e'_h is connected to a neighbor of e_h, and (ii) e'_h is not connected to a neighbor of e_h. If e'_h is connected to a neighbor e_i of e_h, then e_h, e_i and e'_h form an inner C-configuration (see Fig. 9 (i)). This C-configuration must be feasible since e'_h is the highest horizontal edge in the contraction area of C_h.

If e'_h is not connected to a neighbor of e_h then its vertical neighbors must be directed down (see Fig. 9 (ii)). There is only one way–topologically speaking–to connect the neighbors of e_h to the neighbors of e'_h while keeping the polygon

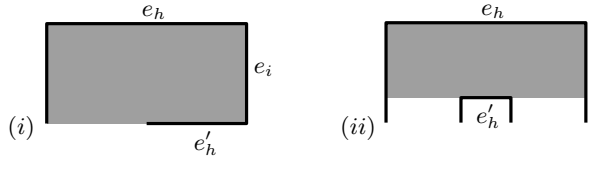


Fig. 9. C_h is not feasible: (i) e'_h is connected to a neighbor of e_h and (ii) e'_h is not connected to a neighbor of e_h

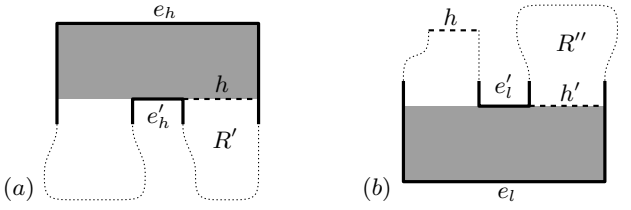


Fig. 10. An example of a virtual edge v completing the right subpart

simple, ensuring that e_h is the highest horizontal edge, and ensuring that e'_h is the highest horizontal edge in the contraction area of C_h (see Fig. 10 (a)). We connect e'_h horizontally to a neighbor of e_h with a *virtual edge* h (see Fig. 10 (a)). The virtual edge splits R into two sub-polygons, both of which have strictly smaller complexity than R. We recurse on the sub-polygon R' that does not contain e_h. We will apply the same reasoning as before, but this time using the lowest horizontal e_l for our argument to avoid h.

Let e_l hence be the lowest horizontal edge of R'. By construction e_l cannot be h. Let C_l be the C-configuration with middle edge e_l. If C_l is feasible, then we are done. Otherwise, if R' is a rectangle, then the only edge inside the contraction area of C_l is h and hence C_l is feasible. It remains to consider the case where R' is not a rectangle and C_l is not feasible. We distinguish two cases (i) and (ii) as above. If we are in case (ii) then we split R' with a virtual edge h' that connects the lowest horizontal edge e'_l inside C_l to one of the neighbors of e_l. Now, and in all further recursions, we have to carefully choose to which neighbor of e_l to connect: we need to create a new sub-polygon R'' that contains neither e_l nor h (see Fig. 10 (b)). There is always exactly one possibility for h'. In R'' we continue again with the highest horizontal edge, and so on. Since the complexity of the polygon under consideration strictly decreases with each iteration, and since the lowest (or highest, depending on parity) C-configuration is always feasible if we recurse down to a rectangle, the lemma follows. □

The *slab* of an edge e, denoted by slab(e), is the region bounded by e and two half-lines orthogonal to e starting at the two endpoints of e, such that the interior of the polygon R does not intersect slab(e) in the immediate neighborhood of e.

Lemma 2. *If there is an edge e such that slab(e) contains a point of the boundary of R which is not directly connected to e, then R has an outer C-configuration.*

Proof. Without loss of generality assume that e is horizontal and that slab(e) lies above e. Let p be the point in slab(e) which is closest to e and not connected to e by a single vertical edge. Further, let q be the closest point to p on e. We connect p and q with a virtual edge h which splits the outside of R into two parts, one bounded and one unbounded (see Fig. 11). Some special care has to be taken if p lies on the boundary of slab(e): if the neighbor e' of e below p is directed upwards, then h connects to the upper vertex of e'.

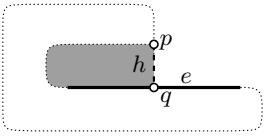


Fig. 11. Finding an outer C-configuration

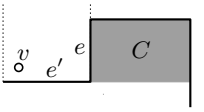


Fig. 12. S-configuration next to C

Denote the bounded part of the outside of R by R' and assume that R' lies to the left of h as depicted in Fig. 11. Let e_l be a leftmost vertical edge of R'. Since e_l is leftmost it must be the middle edge of an outer C-configuration of R. By construction, the virtual edge h cannot be part of this C-configuration. If R' lies to the right of h the argument is symmetric. □

Lemma 3. *Every rectilinear polygon R with at least six edges has either a feasible S-configuration or an outer C-configuration.*

Proof. R has a feasible inner C-configuration C by Lemma 1. Let e be the shorter of the two outer edges of C. Since C is feasible, e is the middle edge of an S-configuration S. If S is feasible, then we are done. Otherwise, denote with e' the neighbor of e that is not part of C. Since S is not feasible, there must be a vertex v of R inside its contraction area. Vertex v cannot lie inside the contraction area of C and must hence lie in the slab of e' (see Fig. 12). Lemma 2 now implies an outer C-configuration. □

Lemma 4. *If a rectilinear polygon R has an outer C-configuration, then it also has a feasible outer C-configuration.*

Proof. The proof of Lemma 4 is very similar to the proof of Lemma 1, so we only sketch the argument. Let C be an outer C-configuration of R. If C is feasible, then we are done. Otherwise let e be the middle edge of C and assume without loss of generality that e is horizontal. Further, assume that R lies locally below e. Let e' be the lowest horizontal edge in the contraction area of C. As before, either e' is directly connected to e and forms a feasible outer C-configuration with their joint neighbor, or e' is not directly connected to e and we can split the outside of R with a virtual edge h into two parts, one bounded and one unbounded (see Fig. 13). We now turn the problem inside out and find a feasible inner C-configuration inside the bounded part of the outside of R. □

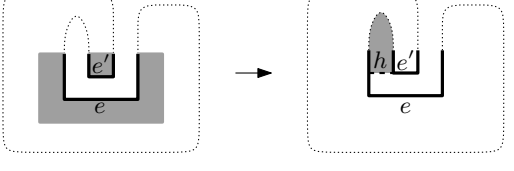


Fig. 13. Finding an inner C-configuration on the outside of R

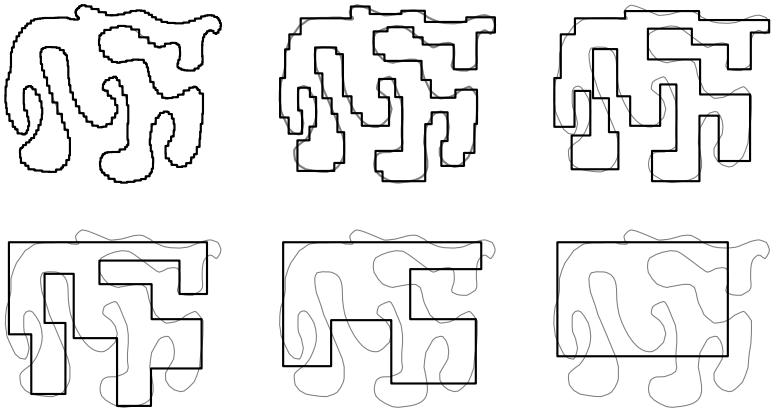


Fig. 14. Schematizing a "Matisse leaf": rectilinearization (with $\alpha = 0.1$), 100, 50, 24, 12, and 4 edges

Theorem 1. *Given a rectilinear polygon R with n edges and an integer k with $4 \leq k \leq n$, an area-preserving schematization of R with at most k edges can be generated using only S- and C-contractions.*

Theorem 1 guarantees that we can always find a feasible contraction, as long as the polygon is not a rectangle. Often we even have a choice out of several feasible operations. Our algorithm then contracts those parts of the subdivision where the contraction results in the smallest symmetric difference with respect to the current schematization (see Fig. 14). C-contractions are inherently not local, the inner and outer C-configurations might lie in completely different parts of the polygon. This can pose a problem with symmetric polygons, insofar that we do not necessarily create symmetric output. We therefore try to find complementary C-configurations which are "close".

To increase the number of feasible contractions, we use a weakened definition of "feasible" for C-contractions. We do not require the contraction area of the larger C-configuration to be empty, it is sufficient if a contraction up to the area of the smaller C-configuration is possible. The bookkeeping necessary for an efficient implementation of our algorithm can be done with standard data structures. Each contraction (re)moves only a constant number of vertices and changes the contraction area of a constant number of edges. This leads to a total running time of $O(n^2)$. Similar to [14] we can easily extend our algorithm to support *landmarks*: special points that lie inside a particular face of the input, must remain in this face, and cannot be moved.

3.2 Subdivisions

Our area-preserving contraction operations can be adapted to schematize subdivisions. If the input subdivision is not rectilinear, then we first use the approach described in Section 2 to turn the input into a rectilinear subdivision. As before,

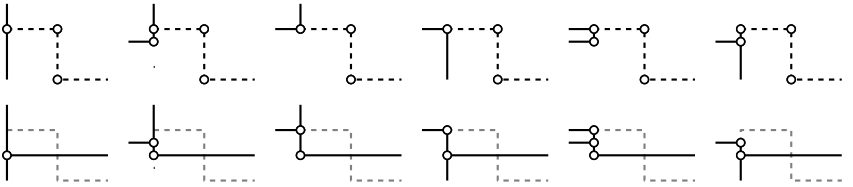


Fig. 15. Several cases for an S-contraction close to a vertex of degree three

we assume that we have only vertices of degree two and three. For subdivisions we cannot guarantee that the schematization can proceed to remove edges until only one edge per polygonal chain remains. However, experimental results show that we can reduce the complexity of the subdivision significantly.

Our ability to use C-contractions in subdivisions is very restricted, we require the middle edges of two complementary C-configurations to be adjacent to the same face of the subdivision. To give the algorithm sufficient flexibility, we allow S-contractions to change edge orientations around vertices of degree three (see Fig. 15). These special S-contractions might remove only one or even zero edges, but are necessary to make further progress afterwards.

4 Distance Measures

In this section we consider the quality of area-preserving approximations of simple rectilinear polygons with respect to three standard distance metrics, namely the symmetric difference, the Hausdorff-distance, and the Fréchet-distance. We show examples for each of these distance measures where among two approximations with equal complexity the one with smaller distance has a worse shape compared to the input polygon. Hence, while it is in principle desirable that an approximation or schematization has a small distance to the input, it is not true that the approximation that minimizes this distance preserves the shape best.

Symmetric difference. The *symmetric difference* between two polygons is defined as the total area that is covered by one polygon but not by the other: it is

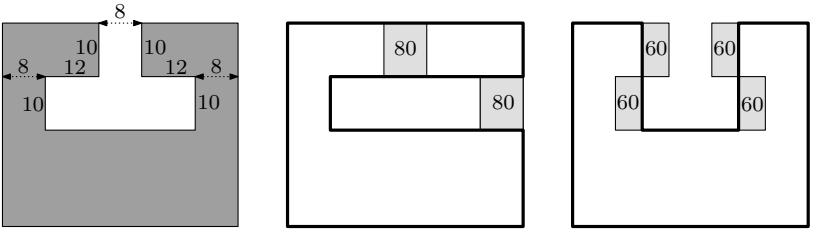


Fig. 16. A polygon and two area-preserving approximations: with minimal symmetric difference (middle), and with better shape (right)

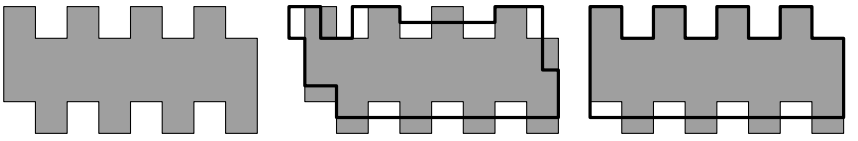


Fig. 17. A polygon and two area-preserving approximations with the same Hausdorff-distance, but significantly different shapes

exactly the area in which they differ from each other. Consider the example in Fig. 16. The input is a 12-sided rectilinear polygon which we would like to approximate in an area-preserving manner with an 8-sided rectilinear polygon. The solution with minimal symmetric difference loses the vertical axis of symmetry and converts the polygon from a U-shape to a C-shape.

Hausdorff-distance. The *Hausdorff-distance* $d_H(X, Y)$ measures the distance between two subsets X and Y of the plane. It finds for each point in X the closest point in Y, and vice versa, and then takes the maximum of these distances. The example in Fig. 16 also shows that an area-preserving approximation with the smallest Hausdorff-distance can have a worse shape than an approximation with a slightly larger distance. Furthermore, since the Hausdorff-distance is determined by a maximum value, the quality of the shape of two area-preserving approximations with the same complexity and Hausdorff-distance can differ greatly (see Fig. 17).

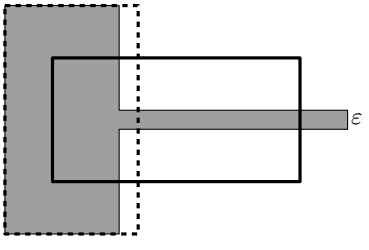


Fig. 18. A polygon and two area-preserving approximations: with minimal Fréchet-distance (solid), and with a better shape (dashed)

Fréchet-distance. The *Fréchet-distance* measures the similarity of two curves by measuring the minimal maximal difference when "walking" along the two curves without moving backwards. Although the Fréchet-distance is defined on continuous curves it can also be applied to polygons. The Fréchet-distance is very sensitive to outliers. Consider the example in Fig. 18 where an 8-sided polygon is approximated by an area-equivalent rectangle. The solution with the smallest Fréchet-distance tries to approximate the thin part, while ignoring most of the polygon.

5 Experimental Results

We have implemented our area-preserving subdivision schematization algorithm and we have generated schematized versions of various polygons and polygonal

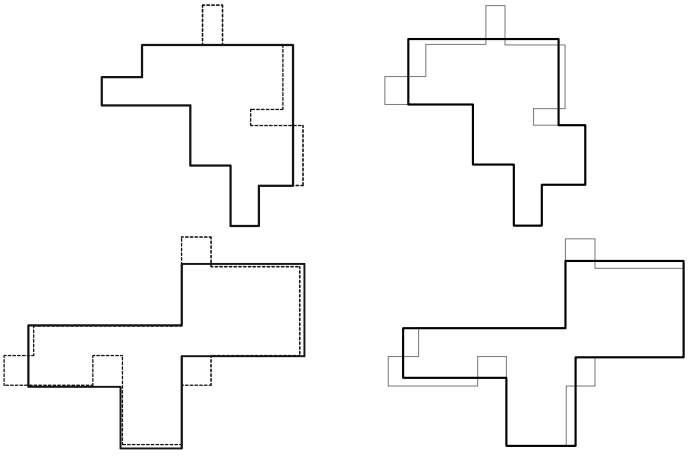


Fig. 19. Two examples from [8] and our schematizations of the same buildings (right)

subdivisions. Fig. 2, 6, and 14 in earlier sections of this paper have all been created by our program. In this section we showcase some additional results.

As mentioned before, our algorithm can also be used to perform building generalization. In Fig. 19 we compare our generalizations to those obtained in [8]. Fig. 20 shows the generalized outline of a castle. In both cases it appears that the simple requirement of area preservation coupled with a greedy approach to minimize the symmetric difference enables us to capture the essential structure of the buildings. The final two figures, Fig. 21 and Fig. 22, show two large subdivisions, namely the provinces and islands of the Netherlands and the countries of Europe, including large lakes and islands. Clearly such maps will benefit from an extension of our methods to the four main orientations, but nevertheless, even the current axis-aligned approach already leads to visually pleasing results.

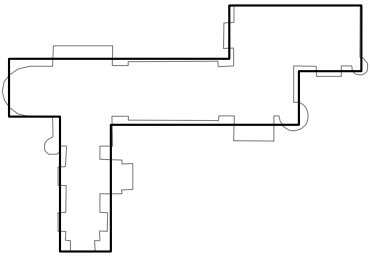


Fig. 20. A schematized castle

6 Conclusions and Open Problems

We described an area-preserving subdivision schematization algorithm which is based on two simple area-preserving contraction operations. We proved that at least one of our operations can always be applied to any given simple rectilinear

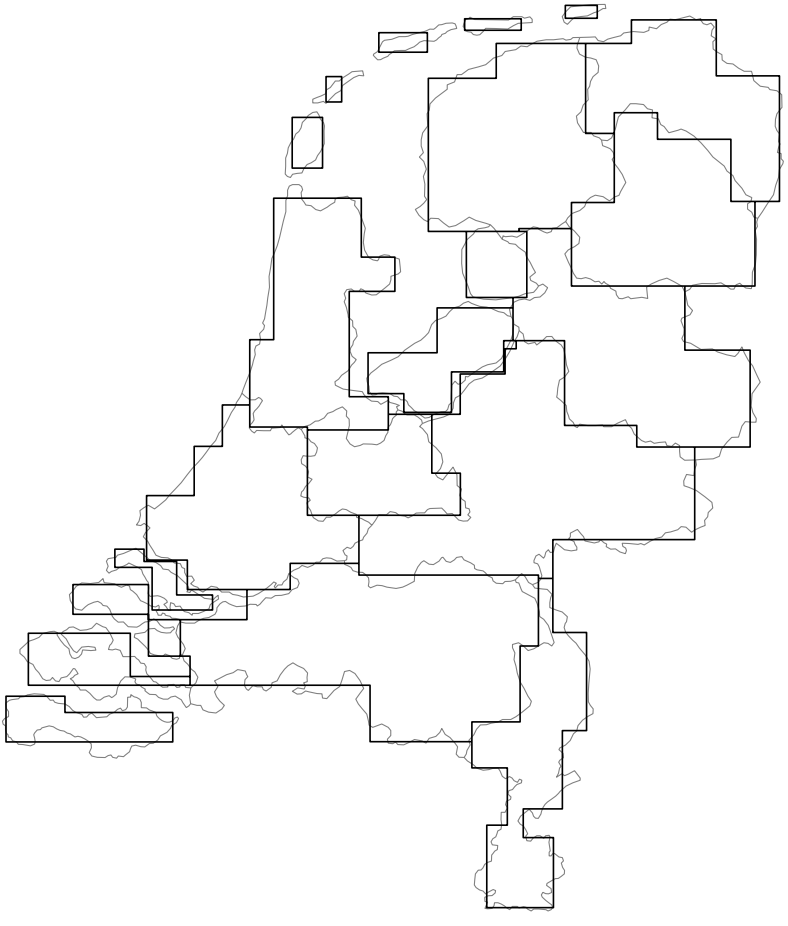


Fig. 21. The provinces and islands of the Netherlands (184 edges)

polygon with at least six vertices. We extended this approach to subdivisions and experimentally evaluated the quality of the resulting schematizations. We also gave examples for standard distance metrics that show that better schematizations might result in worse shapes.

An obvious direction for further work is an extension to the four main orientations. Also, we greedily choose the next contraction that incurs the least symmetric difference. This might lead to an asymmetric schematization of a symmetric polygon and hence other criteria might be more appropriate. Finally, the use of C-contractions is currently very restricted in subdivisions, we require the middle edges of two complementary C-configurations to be adjacent to the same face of the subdivision. Approaches where area is moved via a cycle of neighboring countries might give better results.

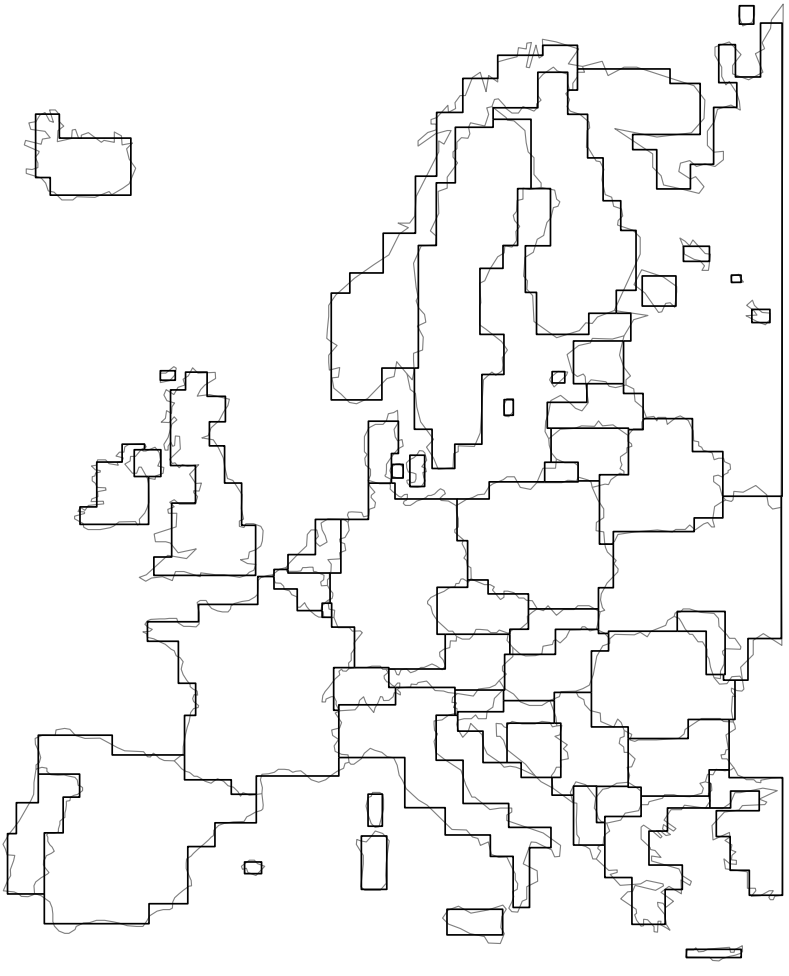


Fig. 22. The countries of Europe, including major islands and lakes (488 edges)

Acknowledgements. The authors would like to thank Marc van Kreveld for helpful discussions on the topic of this paper.

References

1. Cabello, S., de Berg, M., van Kreveld, M.: Schematization of networks. Computational Geometry: Theory and Applications 30(3), 223–238 (2005)
2. Nöllenburg, M., Wolff, A.: A mixed-integer program for drawing high-quality metro maps. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 321–333. Springer, Heidelberg (2006)

3. Swan, J., Anand, S., Ware, M., Jackson, M.: Automated schematization for web service applications. In: Ware, J.M., Taylor, G.E. (eds.) W2GIS 2007. LNCS, vol. 4857, pp. 216–226. Springer, Heidelberg (2007)
4. Damen, J., van Kreveld, M., Spaan, B.: High quality building generalization by extending morphological operators. In: 11th ICA Workshop on Generalization and Multiple Representation, Montpellier, France (2008)
5. Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., Weibel, R.: The application of agents in automated map generalisation. In: 19th International Cartographic Conference (1999)
6. Sester, M.: Generalization based on least squares adjustment. ISPRS - International Archives of Photogrammetry and Remote Sensing 13, 931–938 (2000)
7. Yan, H., Weibel, R., Yang, B.: A multi-parameter approach to automated building grouping and generalization. GeoInformatica 12, 73–89 (2008)
8. Mayer, H.: Scale-spaces for generalization of 3d buildings. International Journal of Geographical Information Science 19, 975–997 (2005)
9. Regnauld, N., Edwardes, A., Barrault, M.: Strategies in building generalisation: modelling the sequence, constraining the choice. In: ICA Workshop on Progress in Automated Map Generalization, Ottawa, Canada (1999)
10. Ruas, A.: Modèle de généralisation de données géographiques à base de contraintes et d'autonomie. PhD thesis, Université de Marne la Vallée (1999)
11. Haunert, J.H., Wolff, A.: Optimal simplification of building ground plans. In: Proceedings of XXIst ISPRS Congress Beijing 2008. IAPRS, vol. XXXVII (Part B2), pp. 372–378 (2008)
12. Sester, M.: Optimization approaches for generalization and data abstraction. International Journal of Geographical Information Science 19, 871–897 (2005)
13. Bose, P., Cabello, S., Cheong, O., Gudmundsson, J., van Kreveld, M., Speckmann, B.: Area-preserving approximations of polygonal paths. Journal of Discrete Algorithms 4(4), 554–566 (2006)
14. de Berg, M., van Kreveld, M., Schirra, S.: A new approach to subdivision simplification. Proceedings of Auto-Carto 12, 79–88 (1995)
15. de Berg, M., van Kreveld, M., Schirra, S.: Topologically correct subdivision simplification using the bandwidth criterion. Cartography and Geographic Information Science 25(4), 243–257 (1998)
16. Estkowski, R., Mitchell, J.S.B.: Simplifying a polygonal subdivision while keeping it simple. In: 17th Symposium on Computational Geometry, pp. 40–49 (2001)
17. van de Kraats, B., van Kreveld, M., Overmars, M.: Printed circuit board simplification: simplifying subdivisions in practice. In: 11th Symposium on Computational Geometry, pp. 430–431 (1995)
18. Guibas, L., Hershberger, J., Mitchell, J., Snoeyink, J.: Approximating polygons and subdivisions with minimum-link paths. International Journal of Computational Geometry and Applications 3, 383–415 (1993)

Periodic Multi-labeling of Public Transit Lines

Valentin Polishchuk and Arto Vihavainen

Helsinki Institute for Information Technology
CS Department, University of Helsinki
P.O. Box 68, FI-00014, University of Helsinki, Finland
{firstname.lastname}@cs.helsinki.fi

Abstract. We designed and implemented a simple and fast heuristic for placing multiple labels along edges of a planar network. As a testbed, real-world data from Google Transit is taken: our implementation outputs an overlay onto Google Maps, adding route numbers to public transit lines.

Keywords: Map labeling, Mass transit lines, Map readability.

1 Introduction

Map labeling comes in many flavors depending, in particular, on the "dimensionality" of the objects to be labeled. Cities, buildings, and other places of interest on a map are "zero-dimensional" points; the requirement to add labels close to them highly restricts the label placement options. Streets, rivers, etc. are one-dimensional; their labels may be placed anywhere on or along them. Countries, states, districts are two-dimensional, and bear labels inside them. A common (often, implicit) assumption in map labeling applications is that the *objects* that have to be labeled are *disjoint*. (Of course, the labels must be disjoint too, and this is the essence of the labeling challenge.)

In this paper, we consider labeling linear features that may *partially overlap*. Hence, parts of the features have to be labeled with multiple labels. Moreover, the features are *long*, which requires that every feature is labeled multiple times, on a periodic basis.

Motivation. Public transportation maps have two formats. *Static* maps are printed in booklets or hung at the stops. To the best of our knowledge, the current practice is to place route labels on these maps manually, which is a rewarding but tedious job. We set out to ease it by automating the label placement. Our implementation allows one to customize the parameters of the label placement procedure, as well as to manually fine-tune the output by adding/moving/deleting/resizing the placed labels.

Recent years have seen a shift towards *online, interactive* maps for journey planning, in which the output content varies on per-query basis, and the product is delivered to end users without involving a human-in-the-loop checking the output for coherence. The possibly leading system here is Google Transit™; many cities employ planners of their own. While these systems work very well for

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 175–188, 2010.

suggesting a path *from A to B*, the possibility of seeing a map of a location with *all* nearby public transport routes labeled, is often missing (the only available option is usually just to zoom on a static map). In many of the systems, one can only see which streets have public transport on them, but unfortunately not the line numbers (Figs. 1 and 2). Even a brute-force solution of putting the labels, say, at the stops would be an improvement; a more intelligent route labeling could be even more helpful.

Related work. Two variations of linear-features labeling have been considered in the literature: *street labeling* and *river labeling* [14, 15, 16, 19, 20]; see also Chapter 58.3.1 in the book [11]. Our problem is different from these because different rivers do not share parts of streambeds, as well as different streets very rarely share the same road (and when they do, it does create confusion for the map reader). On the contrary, in our setting exactly the opposite is the case – many lines may run along the same road (and hence, for instance, a simple solution such as coloring the routes differently may not easily help). The other, more subtle difference is that street and river names are usually long (words) while our labels are generally short (numbers).

In *boundary labeling* the goal is to connect point features inside a region to the labels placed on the region's boundary [4, 5, 6, 7, 13]. While in standard point-feature labeling the label for a feature has only few candidate positions, in boundary labeling labels can be slid arbitrarily along the boundary. Sliding labels was studied in [10, 18, 20], and is also relevant for us, since route labels can be slid along the roads.

For labeling *points*, simple heuristics are known to work well in practice [11, Ch. 58.3.1]. Our heuristic is a practical one for labeling (possibly overlapping) *linear features* with a large number of regularly spaced small-size labels. An automated system for generating an attractions map is presented in [12].

Our contribution. We developed a simple and efficient label generator for routes in a transportation network. To the best of our knowledge, it has no competitors — we are not aware of any automatic system for periodic multi-labeling of overlapping linear features. The latest version of our software is publicly available from `http://www.cs.helsinki.fi/group/compgeom/maplab/`.

As a test case we use real-world public transport route data from Google Transit Data Feed [1]. Sample outputs of our algorithm are presented in figures throughout the paper; Table 1 provides links to online versions of the maps in the figures.

Naturally, the (quality of the) algorithm's output is greatly influenced by the parameter values with which the algorithm is run. The overall number of parameters is small, and a user can familiarize himself with them quickly. However, for the purposes of fully automated label generation (say, when the algorithm is deployed in an interactive online system), no parameter adjustment should be expected from the user. That is, the final, good-looking output must be produced with *default* parameters values. We were able to set the default values so that satisfactory

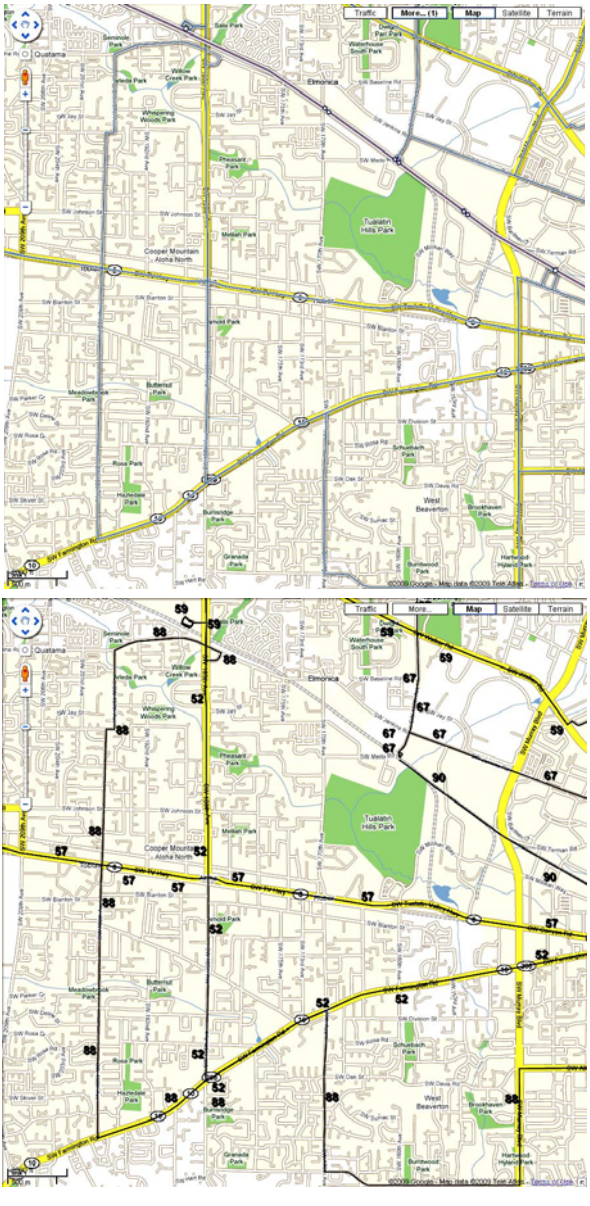


Fig. 1. Top: A snapshot of Google Maps with the Transit overlay; seeing just the routes is not very helpful without knowing which numbers they are. Bottom: Our output overlayed on the same location.

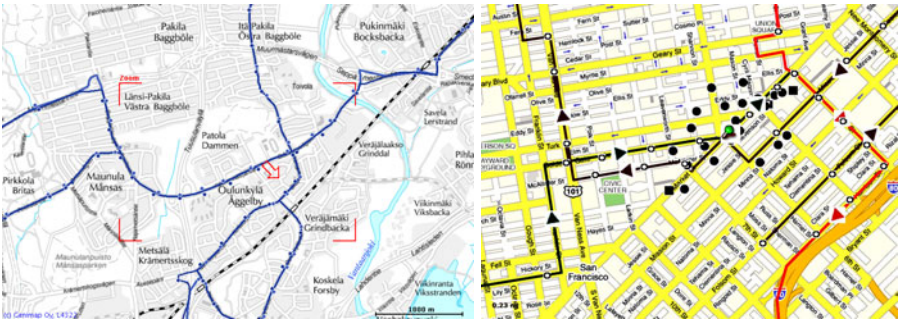


Fig. 2. Cities' journey planners have a functionality to show public transport routes near a location; displaying route labels here could be a huge plus. Left: Helsinki (reittiopas.fi). Right: San Franciso (transit.511.org).

results are obtained without any finetuning. Figures 3 and 6 compare our output with "official" maps produced "once-and-forever" with human oversight.

1.1 Overview of the Approach

The input to our algorithm is a route network – a collection of polygonal paths, each representing a public transit line (Fig. 4, left). The algorithm places route labels along each path, based on a set of user-defined parameters.

Our first step is to preprocess the routes, breaking them into (maximal) stretches such that along each stretch, the set of lines using the stretch is the same. This implies, of course, that along each stretch, the route labels stay the same. So for each stretch, we group the labels of its routes into a "multi-label" – a box that contains the labels of all the lines using the stretch. (Such label grouping is very common in the existing public transport maps of large cities.)

The stretches are further broken, by the intersection points between them, into *subpaths*. This reduces our original labeling problem to the one in which we are given a set of subpaths (pairwise-disjoint other than at endpoints), each with its own multi-label. Our goal now is to place the multi-labels along the subpaths so that the routes are "easy to follow", while satisfying the constraints that the multi-labels stay pairwise-disjoint and do not overlap with any of the subpaths. That is, the boxes corresponding to different multi-labels must not intersect between themselves and, in addition, no box should intersect any of the routes. Satisfying the constraints is necessary and sufficient to guarantee overall clarity of the labeling (we add a small margin to each box in order to enforce some minimum separation between different multi-labels, as well as between a multi-label and a route).

Key Idea: "Propagation" from Intersections

The crux of our approach is the observation that for routes to be easily tracked, it is most important to place route labels near points of routes intersection.

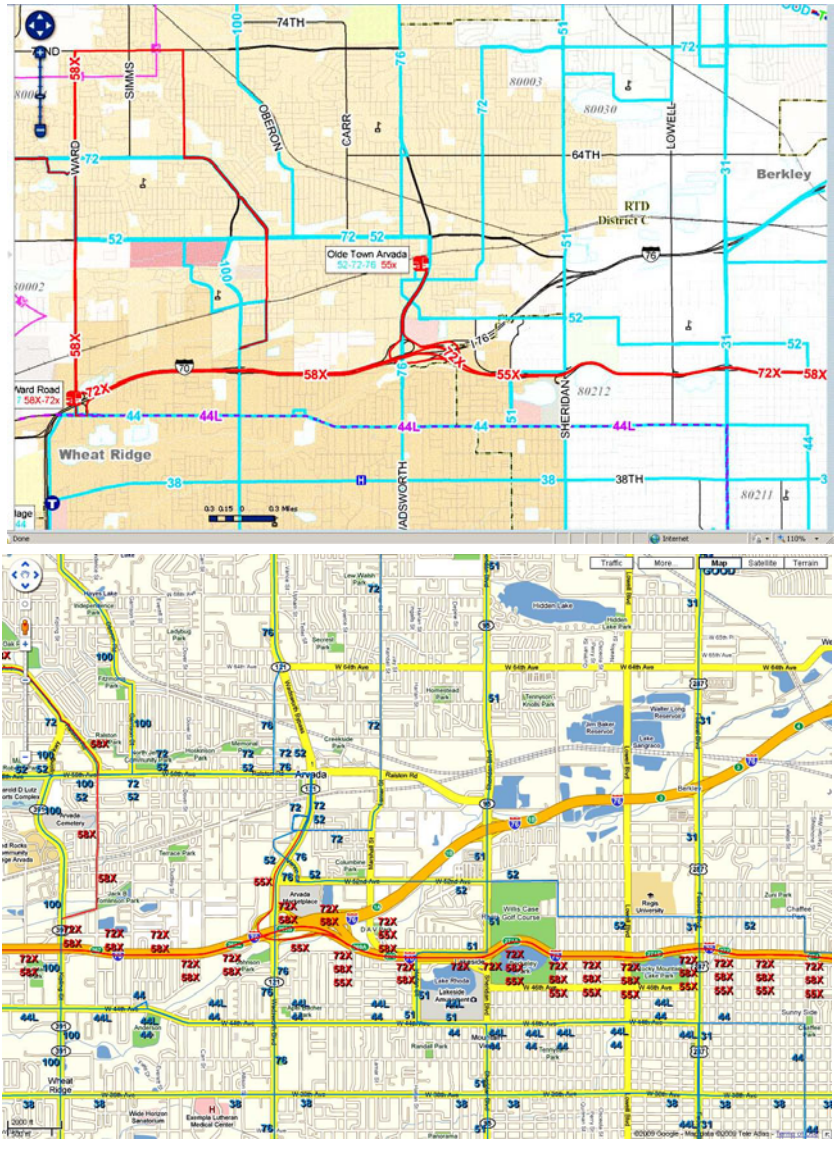


Fig. 3. Official map from rtd-denver.com (top) against our output (bottom).

The intuition is that in a subpath, along which the routes do not change, any particular route can be traced easily. It is only at an intersection point, where the routes diverge and merge, that the route may be lost. Thus, our general approach is as follows:

Step 1: *Place multi-labels near endpoints of subpaths,* and then,
Step 2: *For each subpath, long enough to hold more multi-labels, place them evenly along the subpath.*

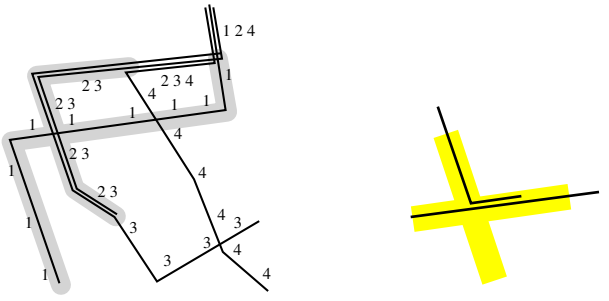


Fig. 4. Left: A network; two of the stretches are enclosed in shaded tubes. Right: Routes merge at an intersection of roads, but the corresponding paths do not cross.

Inside each of the steps, the subpaths are processed in the order of increasing "capacity", which is the maximum number of multi-labels that can be placed along the subpath. The rationale is simple: if subpaths "compete" for a candidate multi-label location, less-capacitated subpaths should be given priority, for if a smaller-capacity subpath is not labeled at the location, it may not get a chance to be labeled at all, while a higher-capacity subpath is more likely to get its multi-label somewhere else along its stretch.

2 Algorithmic Details

We now elaborate on the preprocessing and the label placement steps, and discuss the choice of default values for the parameters.

2.1 Preprocessing

Real-world routes are stored in Google Transit Data Feeds with too high a precision, making the data files unnecessarily large. We simplified the input using Douglas-Peucker algorithm [9].

Another issue is that in the raw data downloadable from Google Transit, different routes going along the same road are often represented by slightly different vertex sequences. This may be due to the fact that the data for different routes comes from different authorities or just because different routes were "sampled" differently. In addition, when two routes merge at an intersection of two roads, the paths that represent the routes may not necessarily intersect because the data is "too precise" (Fig. 4, right).

To address the above issue, we introduce a parameter *mergeThreshold*. Whenever there is a vertex of a route that is closer than the threshold to another route, we snap the vertex to the latter route. Such snapping is implemented as a part of the sweepline algorithm [8, Ch. 2] for detecting intersections between routes; this way the merging does not (asymptotically) increase the running time of the algorithm.

Overall, on the sweep completion we have a set of subpaths, pairwise-disjoint other than at endpoints; each subpath has a unique set of routes that use it. The last step of the preprocessing is creating a multi-label box for each subpath. In case the multi-label contains more than one number (i.e., the subpath is shared by several routes), we arrange the labels into one of the pre-defined rectangular grid-like patterns, and choose the pattern with the smallest area (in case of ties – one with the smallest aspect ratio). We emphasize that the grid is only used for creating multi-labels, not to guide the label placement; the labels can be placed anywhere in the plane.

Note that although our algorithm works with the merged routes, when overlaying the algorithm's output onto Google Maps, we show the original, unmodified routes. This does not hurt the visual effect of label placement as the labels are placed close also to the original routes (and our modification does not distort the routes by much). At the same time, it allows to output all features present in Google Transit data. For example, if routes are colored in the input data, we show them such on the map.

2.2 Label Placement

In what follows, where it causes no confusion, we will call the multi-labels just *labels* (even though a multi-label may contain labels for several lines). We will call the subpaths just *paths*.

For every path we estimate its capacity as follows:

$$capacity \quad \leftarrow \quad 1 + \left\lfloor \frac{pathLength - labelLength - 2 \cdot offset}{labelLength + gap} \right\rfloor \tag{1}$$

where *pathLength* is the length of the path, *labelLength* is the length of the box of the path's multi-label, *offset* is the distance between the label and path endpoint, and *gap* is the gap between the consecutive labels – the last two are user-defined parameters. Of course, the formula is exact only if the path is a horizontal straight-line segment; however, in practice, it gives a good estimate of the number of labels that can be placed along any path.[1]

We first place labels near paths endpoints, processing paths in order of increasing capacity; during the placement, some paths are marked with an 'X' indicating that we give up placing more labels for them:

- Capacity-0 paths are not labeled and are marked with X. Such paths are very short (shorter than the label length), so the user cannot loose a route while following such a path.

[1] This may have several plausible explanations: (1) The majority of the subpaths are straight, and even if not, they "curve" only gently. (2) The subpaths never "zigzag back and forth" so the length of the path reflects well the number of labels that have to be evenly spaced along the path. (3) The boxes have small aspect ratio, so the estimate is not hurt much when a path is not horizontal. (4) Often, the paths are almost perfectly North-South or East-West. (This is common at least in the US cities.)

• For every capacity-1 path, we try to place the label in its midpoint, at distance *roadDistance* from the path where *roadDistance* is a user-specified parameter. If it is not possible to place the label on one side of the path, the other side is tried. After that, whether we succeeded or not, we mark the path with an X: even if the labeling did not succeed, we will not attempt to label the path in the future because capacity-1 paths are still short.
• For $c = 2, 3, \ldots$, for every capacity-c path we try to place the labels near each endpoint of the path, more specifically – at distance *offset* from the endpoint and at distance *roadDistance* from the path. If the placement is not possible, the label is dragged away from the endpoint until a feasible placement is found. Then, another side of the path is tried, and the label remains only on the side for which the dragged distance was smaller.

If no placement is found on either side, the path is marked with X. If the labels dragged from different endpoints come closer than $labelLength + 2{*}gap$, only one label is kept, and the path is marked with X.

Now, for each path not marked with X, we reestimate its "residual" capacity taking into account the locations of its two placed labels (note that only the paths with two labels remain not marked with X):

$$capacity \leftarrow 1 + \left\lfloor \frac{|pathLength(label2) - pathLength(label1)| - labelLength - gap}{labelLength + gap} \right\rfloor \quad (2)$$

where $|pathLength(label2) - pathLength(label1)|$ gives the distance between the placed labels. We place additional labels, evenly spaced on each path. The paths are, again, processed in order of increasing capacity. As before, we try both sides of the path when placing a label, and we drag the label until a feasible placement is found (this time though we do not drag further then for a distance *maxDrag* – another user-specified parameter).

Size matters. In a dense part of the network there may not be enough space to place even a single label of desired size without intersecting the routes. In such situation, it makes sense to reduce label size by using a smaller font for label text. We use a local network complexity metric to determine the areas where the label font size is halved. The complexity metric is based on the total length and the number of routes that intersect a given cell of a regular grid. The complexity of the cell is calculated as the weighted sum of the total length of the routes in the cell and the number of routes crossing the cell boundary (the weights are user-defined parameters).

We shifted the grid by half the cell length in $\pm x$ and $\pm y$ directions, thus obtaining 4 grids. The label size was halved in an area whenever the area was voted to have complexity higher than 10 in 3 or more of the 4 grids (Fig. 5).

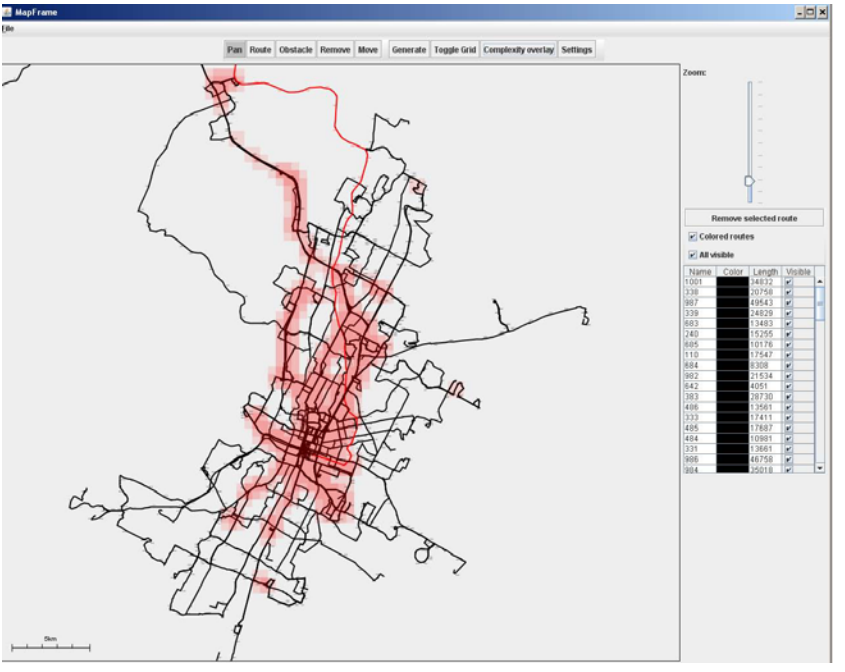


Fig. 5. Complexity overlay over Austin network

2.3 Default Parameters Values

There are three types of parameters:

Map-, or input-related. These are used during the preprocessing (Section 2.1). *DouglasPeuckerEpsilon* is the distance parameter of the Douglas-Peucker line simplification algorithm. Based on the level of detail, visible when viewing the routes at a reasonable zoom (we consider zoom level 12-14 in Google Maps to be "reasonable"), we set $DouglasPeuckerEpsilon = 2$ meters. In most cases, the difference between the original and the simplified routes cannot be noticed by eye while the labeling speeds up substantially.

The *mergeThreshold* defines when two routes are assumed to belong to the same road (see Section 2.1). We set the default value of the parameter to 8 meters based on a typical width of a road.

Complexity-related. The default values $lengthMultiplier = 1.35$, $routeCountMultiplier = .45$ (Section 2.2) were chosen empirically; the default grid size is 1 km.

Output-related. The parameters *labelFontSize*, *roadDistance*, *offset*, *maxDrag*, *gap* are used in the labeling itself (Section 2.2). We choose *labelFontSize* so that it looks approximately as 12pt font when projected on Google Maps at zoom level 14; the height of a number at that scale is about $h = 90$ meters. The rest of the parameters are set based on h: $roadDistance = h/3 \approx 30$m, $offset = 3h \approx 270$m, $gap = 10h \approx 900$m, $maxDrag = 5h \approx 450$m.

All maps in figures in this paper, as well as those at linked URLs were generated by the algorithm running with the default parameter values and without any postprocessing.

3 Results: Overlaying onto Google Maps

The results of our implementation can be overlayed on Google Maps. The overlay is toggled on by entering the location of a .kml file, produced by the implementation, into Google Maps' search box (the box, into which the address is usually typed). Links to the .kml files from the figures in this paper are given in Table 1.[2]

Table 1. To see the map, click on the link or copy and paste it into Google Maps search box. For better contrast, remove the satellite image in Google Maps (click the Map button).

Figure/ City	Link to the kml file (may be input into Google Maps search box)
Title page	`http://cs.helsinki.fi/~polishch/pages/map/austinPark.kml`
Fig. 1	`http://cs.helsinki.fi/~polishch/pages/map/portlandSmall.kml`
Fig. 3	`http://cs.helsinki.fi/~polishch/pages/map/denverSmall.kml`
Fig. 6	`http://cs.helsinki.fi/~polishch/pages/map/portlandSBS.kml`
Austin	`http://cs.helsinki.fi/~polishch/pages/map/austin.kml`
Dallas	`http://cs.helsinki.fi/~polishch/pages/map/dallas.kml`
Denver	`http://cs.helsinki.fi/~polishch/pages/map/denver.kml`
Portland	`http://cs.helsinki.fi/~polishch/pages/map/portland.kml`

We also generated .kml files with labeled routes for whole cities. Unfortunately, Google Maps limits the size and the complexity of displayed .kml files [2]. Due to the restrictions, not all labels produced by our implementation for the cities are actually rendered on Google Maps. Anyway, in Table 1 we offer links to the cities' .kml files, but note that our output cannot be fairly judged for them because of the many missing labels. (For the maps, presented in the figures, we ran our algorithm on small subsets of routes visible for each figure; thus, all our labels should be seen for them.)

Running times. Table 2 shows the times to generate labels for some cities from Google Transit data feed. The Loading column shows times to load the data from the harddrive, not to download from the data feed. When the data is saved in our internal format, the load times are 2 to 10 times faster.

[2] One and the same city has several data files in Google Transit Data Feed, which differ a lot one from another. We suspect in certain cases none of the files is actually close to reality; in fact, Google Transit directions sometimes use routes that are not in the database.

Table 2. Label generation statistics on Intel Core 2 Duo E6600, 4G RAM, Fedora Core 12, Java 1.6.0_16

City	Filesize kbytes	Loading ms	Generating ms	Number of labels	kml file size, kbytes
Albany, NY	6887	1261	8135	987	569
Arlington, VA	171	121	883	410	188
Denver, CO	53130	14402	15504	4231	2137
Houston, TX	12760	3416	16159	2185	484
Los Angeles, CA	31240	7615	7787	2489	1187
Portland, OR	29713	7001	4669	1590	793
Sacramento, CA	4082	1266	5581	1288	624
Washington, DC	57273	12042	2699	3430	1947

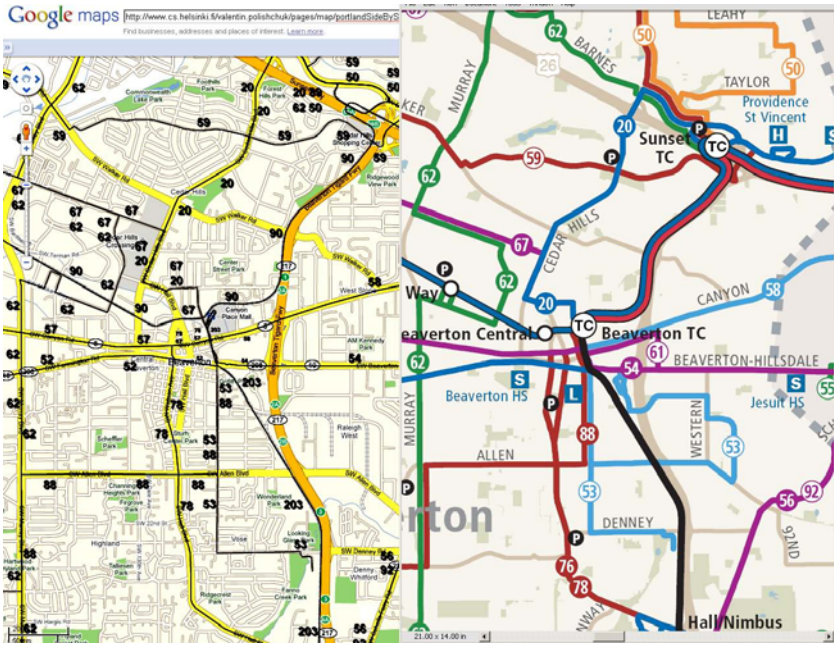


Fig. 6. Somewhere in Portland: our output side-by-side with official map (trimet.org)

4 Discussion

It is easy to add to our implementation the functionalities like using different colors, thickness, style, fonts, etc. for different routes (in fact, the implementation has the possibility to manually change route colors, or to use the colors that come from Google Transit Data Feeds). Our goal though was to keep the output "minimalistic" w.r.t. the number of ways used to distinguish between the routes. Even with that, our map can compete in readability with the ones using the power of coloring, different label styles, and human supervision (see Figs. 3 and 6).

4.1 Limitations of the Algorithm

We now outline several points of possible improvement to our system.

Scaling with zooming. The biggest issue is that our labels are images, and hence there are only a handful of zoom levels at which they look good. For too high or too low zooms, one would have to scale the labels and recompute the periodic distances. Alternatively, our implementation can output text labels; this open up the question of deciding active ranges [3] for the labels. (Taking into account the specifics of our application, one may ask that the labels near intersections have larger ranges.)

Clipping to the visible area. A related issue is that we are labeling the routes for a whole city, while a user may be interested in seeing only a small "window" of the map. Our current implementation is missing the possibility to "clip" the routes to a specified rectangle, and restrict the labeling to it (possibly, adding a margin to enable smooth map panning). The running times in Table 2 hint at the limits of what our approach can do for a whole city.

City centers. As expected, our output is worst in the city centers where the routes form a complicated network, often with almost all lines having a terminal point. A simplest instance when our approach fails is when 3 roads run parallel to each other – then the middle road will not be labeled.

Label-road matching. When a label is placed close to the intersection of two roads, the user may err in deciding to which road the label actually belongs. In such cases, removing such ambiguity is called for.

Other labels. We ignored interaction with other labels present on the map. While street names pose no problem to us because we are placing labels outside the roads, there may be certain places on the map which must not be hidden beneath the route labels.

Complexity definition. Our network complexity computation is somewhat ad-hoc. Are there better ways to capture the essentials of what constitutes a "problematic" area? E.g., one can define complexity based on a quadtree subdivision of the map; will that lead to a better grasp on the local map complexity?

Evaluation. We are unaware of a clear objective criteria to measure how good a route labeling is. One possibility is to evaluate the quality by using the framework of [17] or by running a user study; the latter, in particular, may suggest better choices for the parameters, as well as for the fonts, sizes and styles of the labels.

Acknowledgments

We thank the anonymous referees for their suggestions. We acknowledge discussions with members of Algorithms groups at Stony Brook University, TU

Eindhoven and Karlsruhe Institute of Technology. Initial coding was done by David Consuegra, Vesa Hautsalo, Niko Himanen, Anttijuhani Lantto and Mikko Sysikaski during a class project at CS Department, the University of Helsinki. Google Maps™and Google Transit™are Google Brand Features. This research is partially supported by Academy of Finland grant 118653 (ALGODAN).

References

1. http://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds
2. http://code.google.com/apis/kml/documentation/mapsSupport.html
3. Been, K., Nöllenburg, M., Poon, S.-H., Wolff, A.: Optimizing active ranges for consistent dynamic map labeling. In: SCG 2008: Proceedings of the 24th Annual Symposium on Computational Geometry, pp. 10–19 (2008)
4. Bekos, M.A., Kaufmann, M., Potika, K., Symvonis, A.: Multi-stack boundary labeling problems. In: Arun-Kumar, S., Garg, N. (eds.) FSTTCS 2006. LNCS, vol. 4337, pp. 81–92. Springer, Heidelberg (2006)
5. Bekos, M.A., Kaufmann, M., Symvonis, A., Wolff, A.: Boundary labeling: Models and efficient algorithms for rectangular maps. Comput. Geom. 36(3), 215–236 (2007)
6. Benkert, M., Haverkort, H., Kroll, M., Nöllenburg, M.: Algorithms for multi-criteria boundary labeling. Journal of Graph Algorithms and Applications (2009)
7. Benkert, M., Nöllenburg, M.: Improved algorithms for length-minimal one-sided boundary labeling. In: 23rd European Workshop on Computational Geometry (EuroCG 2007), pp. 190–193 (2007)
8. Berg, M.d., Cheong, O., Kreveld, M.v., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer, Heidelberg (2008)
9. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Canadian Cartographer 10(2), 112–122 (1973)
10. Garrido, M.A., Iturriaga, C., Márquez, A., Portillo, J.R., Reyes, P., Wolff, A.: Labeling subway lines. In: Eades, P., Takaoka, T. (eds.) ISAAC 2001. LNCS, vol. 2223, pp. 649–659. Springer, Heidelberg (2001)
11. Goodman, J.E., O'Rourke, J. (eds.): Handbook of discrete and computational geometry. CRC Press, Inc., Boca Raton (1997)
12. Grabler, F., Agrawala, M., Sumner, R.W., Pauly, M.: Automatic generation of tourist maps. ACM Trans. Graph. 27(3), 1–11 (2008)
13. Iturriaga, C., Lubiw, A.: Elastic labels around the perimeter of a map. J. Algorithms 47(1), 14–39 (2003)
14. Neyer, G., Wagner, F.: Labeling downtown. In: Bongiovanni, G., Petreschi, R., Gambosi, G. (eds.) CIAC 2000. LNCS, vol. 1767, pp. 113–124. Springer, Heidelberg (2000)
15. Seibert, S., Unger, W.: The hardness of placing street names in a manhattan type map. In: Bongiovanni, G., Petreschi, R., Gambosi, G. (eds.) CIAC 2000. LNCS, vol. 1767, pp. 102–112. Springer, Heidelberg (2000)
16. Strijk, T.: Geometric algorithms for cartographic label placement. PhD thesis, Utrecht University (2001)
17. Dijk, S.v., Kreveld, M.v., Strijk, T., Wolff, A.: Towards an evaluation of quality for label placement methods. In: Proceedings of the 19th International Cartographic Conference, Ottawa, International Cartographic Association, pp. 905–913 (1999)

18. Kreveld, M.v., Strijk, T., Wolff, A.: Point labeling with sliding labels. Comput. Geom. Theory Appl. 13(1), 21–47 (1999)
19. Wolff, A., Knipping, L., Kreveld, M.v., Strijk, T., Agarwal, P.K.: A simple and efficient algorithm for high-quality line labeling. In: Atkinson, P.M., Martin, D.J. (eds.) Innovations in GIS VII: GeoComputation, vol. 11, pp. 147–159
20. Yu, K.-L., Liao, C.-S., Lee, D.-T.: Maximizing the number of independent labels in the plane. In: Preparata, F.P., Fang, Q. (eds.) FAW 2007. LNCS, vol. 4613, pp. 136–147. Springer, Heidelberg (2007)

Comparing the Effectiveness of GPS-Enhanced Voice Guidance for Pedestrians with Metric- and Landmark-Based Instruction Sets

Karl Rehrl, Elisabeth Häusler, and Sven Leitinger

Salzburg Research, Jakob Haringer Straße 5/3,
5020 Salzburg, Austria
{karl.rehrl,elisabeth.haeusler}@salzburgresearch.at,
sven.leitinger@salzburgresearch.at

Abstract. This paper reports on a field experiment comparing two different kinds of verbal turn instructions in the context of GPS-based pedestrian navigation. The experiment was conducted in the city of Salzburg with 20 participants. Both instruction sets were based on qualitative turn direction concepts. The first one was enhanced with metric distance information and the second one was enhanced with landmark-anchored directions gathered from participants of a previous field experiment. The results show that in context of GPS-enhanced pedestrian navigation both kinds of instruction sets lead to similar navigation performance. Results also demonstrate that effective voice-only guidance of pedestrians in unfamiliar environments at a minimal error rate and without stopping the walk is feasible. Although both kinds of instructions lead to similar navigation performance, participants clearly preferred landmark-enhanced instructions.

Keywords: Pedestrian navigation, location-based services, cognitively ergonomic turn instructions, navigation performance, voice-only guidance.

1 Introduction

GPS-based electronic navigation aids for supporting people in everyday navigation tasks are becoming more and more popular. Being originally designed for car drivers, attention has recently shifted to pedestrians [1]. While in-car navigation aids have evolved over years, effective GPS-based navigation for pedestrians is still an open issue ([2], [3], [4]). Beside challenges in accurate positioning, one of the most crucial questions concerns the effective communication of route knowledge. Generally spoken, route communication for electronic navigation assistance can either be (1) map-based, (2) text-based, (3) voice-based, (4) tactile, (5) augmented reality overlay and (6) arbitrary combinations of these basic kinds. A number of different choices of route communication have been explored in previous work. The performance of electronic map guidance has been criticized due to problems with the small display-size of electronic devices [5]. Although major improvements like automatic aligning of the map in heading direction are commonly implemented, some other difficulties like stopping the walk for viewing the map or aligning 2D representations to the real world remain

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 189–203, 2010.

([6], [7], [8]). 3D maps, although recently implemented by system vendors, do not lead to considerable improvements [9]. Augmented reality overlays are able to improve the alignment task by augmenting the real world view with virtual route information [10]. Although first attempts proved as useful, due to technical limitations augmented reality navigation has not come to widespread use. Text-based guidance works basically well but also suffers from the drawback that people have to stop their walk to read the texts [7].

Stopping the walk during the navigation task seems to be one of the major drawbacks of many kinds of route communication in the context of pedestrian navigation. Two promising approaches are suited to overcome the problem: tactile [11] and voice-based guidance [12]. Whereas tactile guidance as minimal attention interface has been successfully demonstrated, voice-only guidance for pedestrians is considered to be not very successful. Goodman et al. [7] conclude that pedestrians with voice-only guidance (without GPS support) perform significantly worse than with other kinds of navigation support. Ishikawa et al. [4] confirmed the poor performance of GPS-based navigation support in comparison to maps and direct experience. Although the authors tackled the aspect of GPS positioning, they did not separate voice from map-guidance. Additionally voice instructions were only used infrequently. In subsequent work Ishikawa et al. [13] tested the effects of different frames of reference with voice instructions. Although in this study voice instructions were carefully constructed, the aspect of automatic triggering of instructions as a consequence of GPS-positioning was missing. In contrast to the work of Ishikawa et al. [13], Streeter et al. [14] prove a significantly better navigation performance of car drivers with carefully constructed and taped audio-instructions in comparison to route maps. Goodman et al. [7] make the point that in the context of voice-based pedestrian navigation a number of questions are open for further research (e.g. more effort has to be put in the construction of voice instructions).

Recent work stresses the need for cognitively ergonomic route directions [15] for effective guidance. Klippel et al. [16] contribute to this question with the wayfinding choreme theory, a formally defined model of cognitively ergonomic turn direction concepts. Denis' work [17] contributes user-generated skeletal route descriptions, which were rated best by people in need of assistance. The authors further stress the need for landmarks as key parts of good verbal route descriptions [18]. The importance of using landmarks for anchoring navigation instructions in space is confirmed by several authors ([19], [20]). Since varying definitions of the concept *landmark* exist ([21], [22]) we commit to the notion of *local landmarks* given by Raubal and Winter [23] due to best fit in the context of pedestrian navigation. We use the definition of *local landmarks* as "salient visible features in the environment which can be used for anchoring qualitative spatial actions for the precise description of route segments". Concerning the effects of using local landmarks in pedestrian navigation Ross et al. [24] report positive effects. However, in their experiments instructions were communicated as written text on flip cards. Effects of automatically triggered voice instructions were not tested which gives a strong motivation for further research.

From the review of previous work we conclude the following hypothesis: The effectiveness of voice-only navigation support for pedestrians could be significantly improved with automatically triggered and systematically constructed cognitively ergonomic voice instructions. This hypothesis implies to main research questions:

(1) Which effects on navigation performance can be attributed to the automatic triggering of voice instructions and (2) which effects can be attributed to cognitively ergonomic instructions. For addressing these questions three specific hypotheses are formulated:

(1) Cognitively ergonomic voice instructions given at (potential) decision points allow for an effective navigation of pedestrians (with a minimal error rate) along a pre-defined, unknown route.
(2) With voice-enhanced guidance unknown routes can be navigated in standard walking times (standard walking time means to walk the route at walking speed typical for a specific age group [25]).
(3) The use of landmark-enhanced voice instructions shows a positive effect on navigation performance (in terms of navigation errors and navigation time) as well as user confidence.

To prove these hypotheses a field experiment with 20 participants was conducted. The paper reports on the set-up, the implementation and the results of the experiment. Proving the hypotheses contributes to a better understanding of voice-based pedestrian navigation with GPS-enabled mobile devices and answers some of the questions raised in previous work [7].

The remainder of this paper is organised as follows: Section 2 reports on the set-up and implementation of the field experiment. In Section 3 results of the experiment are reported and discussed. Finally, Section 4 discusses results and gives an outlook on further work.

2 Field Experiment

As already stated in the introduction this research is subsequent to the work reported in [26] and uses the same study area as well as previously gathered user descriptions of route choices.

2.1 Set-Up

Participants. 20 participants were selected for the field experiments (all different to the previous experiment). Their ages were between 19 and 27 years (M = 21.85 years). Participants were half female and half male and got paid for participation. They were first term students staying no longer than three weeks in the city. All participants confirmed to have no or very limited spatial knowledge of the city and to be unfamiliar with the test routes. Each participant had to complete both routes (described below). All participants indicated to be German native speakers.

Study Area and Routes. As study area we selected the same routes in Salzburg as described in the previous experiment [26] (the Vienna routes were not considered for this experiment). The *Inner City Route* is situated in the historical district of Salzburg. The *Lehen Route* is situated in the Lehen district, a residential area 15 minutes away from the inner city.

The Inner City Route is 1.437 meters long, consisting of 25 decision points which were identified by users in the previous experiment. Due to the high agreement of participants on decision points, we did not question the selection, but considered it as

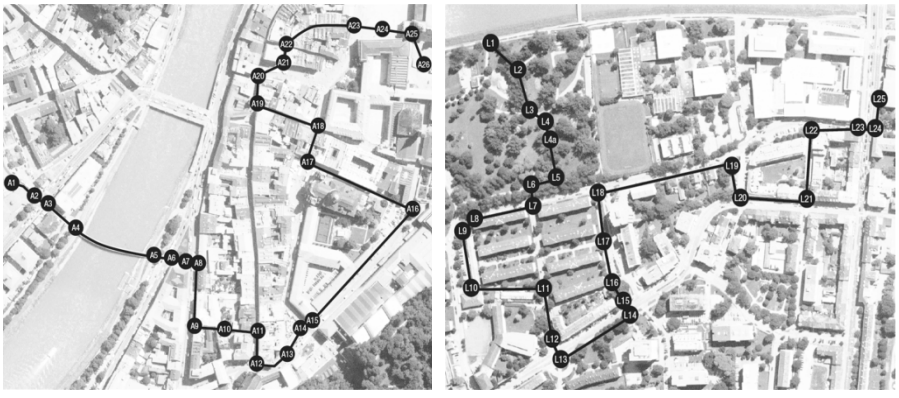


Fig. 1. (1) Inner City Route and (2) Lehen Route with user-defined decision points

user-defined. The Lehen Route is 1.287 meters long with 24 user-defined decision points. In comparison to other experiments [4] these routes are significantly longer with a higher number of different decision situations (staying on track, turning in different directions, and passing different structures of decision points). Fig. 1 shows the two routes and the user-defined decision points on aerial images of the districts.

Voice instructions. We created two sets of German voice instructions. For the *metric instruction set* qualitative turn direction concepts were enhanced with metric distance information. Turn direction concepts were generated by mapping turn angles (between the previous and the following route segments) on wayfinding choremes [27]. Metric distances were constructed by measuring the length of route segments. A typical instruction in the metric instruction set is "Walk straight for 43 meters". Table 1 shows composition principles.

Table 1. Example of composing a metric voice instruction for a route segment

		Action	*Direction/Relation*	*Distance*
Turn Concept	walk straight	walk	straight	
Distance	43 meters	walk		43 meters

For the *landmark instruction set* turn direction concepts were enhanced with qualitative actions describing the motion path along a specific route segment. All qualitative actions were extracted from verbal descriptions given by participants in the previous experiments [26]. The actions for each route segment were modelled with three different sets of particles:

(a) *Motion concepts*: A set of re-occurring verbs describing motion patterns. Frequently used verbs were *turn, walk, pass* and *cross*.
(b) *Direction concepts*: A set of re-occurring spatial relations which can be used to anchor motion with landmarks. Frequently used spatial relations were *along, in, out, direction of, to, through, through between*.

(c) *Landmark concepts*: A set of linguistic concepts used for describing reference entities along different route segments. For each entity type the most frequently used linguistic concept was extracted. Proper names of physical entities were only used in case of clearly readable signs (this naming strategy was used by participants in the previous experiments).

The selection of the specific concepts for each decision point was based on the occurrence frequency in the verbal protocols of the previous experiment. Qualitative actions were *goal actions* (e.g. "Walk to the bridge"), *path actions* (e.g. "Walk through the archway"), *trajectory actions* (e.g. "Walk along the river") or *orientation actions* (e.g. "Walk towards the church"), depending primarily on the used direction concept.

Each instruction of the landmark-based instruction set was composed from exactly one turn direction concept and at least one *goal, path, trajectory* or *orientation* action. No more than three qualitative actions were added to an individual instruction. Table 2 shows the composition of an instruction in the landmark-based instruction set.

Table 2. Example of composing a landmark-based voice instruction for a route segment

		Action	*Direction/Relation*	*Landmark*
Turn Concept	walk straight	walk	straight	
Path Action	pass theatre	pass		theatre
Goal Action	walk to crossing	walk	to	crossing

The complete instruction from the example above is "Walk straight, pass the theatre and walk to the crossing". Instructions were spoken by a skilled speaker (a person working as a teacher for several years) and taped before the experiment in order to avoid any problems with synthesized voice output.

Since both instruction sets use equal turn direction concepts, the varying additional information (metric and landmark-based enhancements) is used to clarify the question, whether turn direction concepts or additional information have the main impact on navigation performance. If navigation performance with both instruction sets is equal, we conclude that mainly turn direction concepts account for navigation performance. If navigation performance is better with one of the instructions sets, we conclude that mainly the specific information enhanced to turn direction concepts accounts for navigation performance.

Self-Report Sense of Direction. To get insights on the spatial abilities of the test group, we used the standardized Santa Barbara Sense-Of-Direction Scale (SBSDS). This questionnaire consisting of fifteen 7-point Likert-type questions with half of the questions stated positively and the other half stated negatively was used for concluding from differences in spatial abilities ([4], [28]).

Quantitative and Qualitative Interviews. After participants finished a route we asked them different quantitative and qualitative questions concerning their personal experiences as well as their confidence during the navigation task. More specifically, we were interested in how they generally felt with GPS-based voice-only guidance and how they felt with the two different kinds of instruction sets. We also asked them whether they were facing any problems in understanding and interpreting the voice

commands. In case of wrong decisions we asked them to tell us their reason for the wrong decision.

2.2 Implementation

The experiment was implemented with two test groups. Participants of each group had to complete both routes. Between the groups we mutually changed the order of instruction sets. One group started each route with *metric instructions* and was switched to *landmark instructions* after completing half of the route. The other group started with *landmark instructions* and was switched to *metric instructions* at halfway. Switching was done at decision point A13 along the Inner City Route and at decision point L12 along the Lehen Route (see also Fig. 1). Switching instructions halfway should avoid learning effects and biases of participants along routes.

All participants started the experiment with the Santa Barbara Sense-of-Direction scale. The results of the test were captured electronically with the MobileSurvey-Suite[1], a mobile application designed for implementing in-situ questionnaires.

Fig. 2. Impressions from the experiment (Inner City Route)

After completing the SBSDS participants were introduced to the experiment. During the navigation task each participant was closely followed by two experimenters, one acting as instructor and one monitoring navigation performance. Participants were equipped with a Nokia N95 mobile phone running the voice-based navigation application and being connected to a head phone. The head phone was completely surrounding the ears in order to prevent street noise from disrupting voice commands. At any time the instructor guaranteed for the safety of participants. In order to provide equal conditions for all participants and to avoid positioning problems coming from distorted GPS signals we decided to simulate GPS-positions by triggering voice commands manually at pre-defined positions. Therefore we used a method called *Wizard-of-Oz Prototyping* which has been proven to be well suited for in-situ experiments [29]. The instructor of the experiment (being the same person during all experiments) was equipped with a second mobile phone which was wirelessly connected to the participant's mobile phone. The experimenter used the second mobile phone to trigger voice commands at pre-defined locations immediately before a decision point. This set-up guaranteed that all participants got the instructions exactly at the same position.

[1] See also https://mobilesurveysuite.salzburgresearch.at

While one experimenter was responsible for triggering voice commands timely, another experimenter documented the results using the MobileSurveySuite. The second experimenter continuously documented participants' decisions, whereas the time needed to walk along a route segment was measured automatically by the application. If a decision was wrong, gestures were used to indicate participants the right choice. No other assistance was given during navigation. In order to avoid any influence on participants, the two experimenters walked a few steps behind the participants. An interview with qualitative questions documented impressions immediately after completing a route. After both routes participants had to complete a standardized questionnaire with twenty quantitative and qualitative questions.

3 Results

3.1 Sense of Direction

For each participant, we calculated the mean values of their answers to the fifteen questions of the SBSDS. Similar to Ishikawa et al. [14] we reversed negatively stated questions to positively stated ones so that a higher score means a better sense of direction. The results of the SBSDS (mean value = 3.89; SD = 0.76) revealed no significant difference in sense-of-direction between the 20 participants. Female participants estimated their sense-of-direction worse than male (Female: 4.09 (0.42), Male: 3.70 (0.99)). Since the calculated mean value in Ishikawa et al. [14] is similar (mean = 3.6; SD = 1.2), we consider our test group as balanced regarding sense-of-direction.

3.2 Walking Times

Walking times of route segments were automatically recorded by the participant's mobile device. Recording was started at the beginning of a voice instruction of the route segment and stopped with the voice instruction for the next segment. From these data we calculated mean walking times of all participants for the first and the second part of each route (routes were split at switching from landmark to metric instructions and vice versa). The two route parts were of different lengths: Inner City Route, Part 1: 594 m, Part 2: 843 m; Lehen Route, Part 1: 693 m, Part 2: 594 m.

In order to prove the second hypothesis we calculated standard walking times for the four route parts with a walking speed of 1.51 m/s, which is according to Knoblauch et al. [25] an empirically founded walking speed for a younger-aged group. For accepting the hypothesis, mean walking times of participants should not be longer than the calculated standard walking times.

Fig. 3 compares standard walking times to the mean walking times of the test group receiving metric instructions and the test group receiving landmark instructions for each of the four route parts. The figure reveals that for two route parts (Route Lehen Part 1 and Route Inner City Part 2) the walking times during the experiment nearly match with standard walking times. On the Route Lehen, Part 1 the mean walking time of participants is 5.61% (SD = 6.48%) lower with landmark instructions and 5.87% (SD = 13.85%) lower with metric instructions compared to standard walking times. On the Route Inner City, Part 2 the mean walking time is 6.94% (SD = 7.16%) higher with landmark instructions and 5.11% (SD = 5.86%) higher with

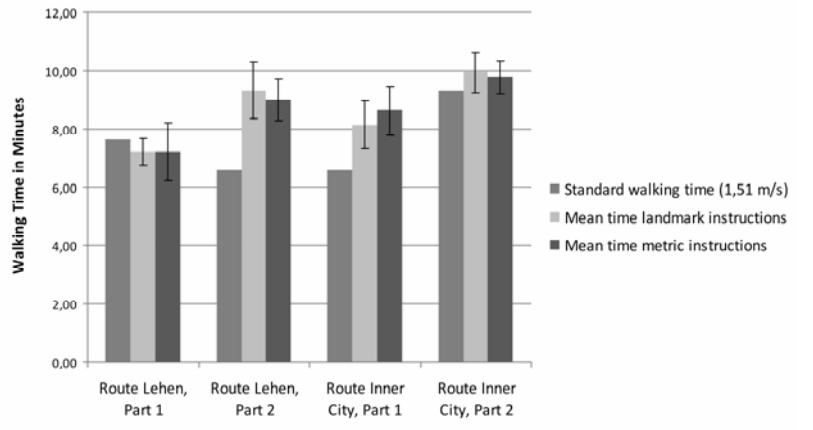


Fig. 3. Mean walking times of participants. Error bars show standard deviations.

metric instructions compared to standard walking times. This result indicates that participants could navigate these route parts nearly in standard walking time without having to stop their walk. However, Fig. 3 reveals contrary results for the two other route parts. Along the Route Lehen, Part 2 the mean walking time of participants is 42.31% (SD = 10.93%) higher with landmark instructions and 37.27% (SD = 7.68%) higher with metric instructions compared to standard walking times. Along the Inner City Route, Part 1 the mean walking time of participants is 24% (SD = 9.46%) higher with landmark instructions and 31.63% (SD = 10.46%) higher with metric instructions compared to standard walking times. These results indicate that the walk had to be stopped during navigation. A detailed analysis of walking times revealed, that the higher walking times along these route parts are originating from a few route segments crossing major streets with intense traffic. Along these route segments (three crossings on each route part, partly equipped with traffic lights) participants sometimes were forced to stop their walk due to red lights. Along the other route parts participants did not have to cross major streets or at least only one time (Inner City Route, Part 2). The average results from Fig. 3 perfectly fit in this picture. Thus we conclude, that increased average walking times are mainly a result of crossing major streets, and not of different instruction sets. Therefore, the hypothesis claiming that test persons being unfamiliar with a navigation environment can be guided along predefined routes at standard walking times can be approved. However, results also reveal that in case of crossing major streets, standard walking times have to be adjusted to more realistic walking times including time for waiting at red lights.

Results in Fig. 3 additionally reveal that different instruction sets (landmark vs. metric) do not have any influence on average walking speed. Due to the continuous location-based communication of instructions and the minimal attention aspect of the voice interface there is no need for people to stop their walk for orientation purposes, which can be seen as strong argument for the effectiveness of the proposed solution.

3.3 Decision Errors

The number of wrong decisions indicates how often participants misinterpreted verbal instructions so that they deviated from the route. We formulated the hypothesis that cognitively ergonomic and location-based voice instructions support people effectively in their navigation task and minimize the risk to deviate from the route.

Fig. 4 reveals a total of 18 wrong decisions. Taking 980 decisions as total (20 participants making decisions at 49 decision points), 18 wrong decisions correspond to a relative error rate of 0.02%. With 99.98% of all possible choices taken correctly the first hypothesis can also be approved. Although the error rate is extremely low with both instruction sets, Fig. 4 reveals that metric instructions resulted in a higher error rate compared to landmark instructions. Comparing routes, navigation along the Lehen Route resulted in a higher error rate compared to the Inner City Route. The Inner City Route with landmark instructions was the only route completed without errors by all participants.

Wrong decisions along the Lehen Route mainly occurred at decision points L4 and L4a (which were decision points within a park area, see also Fig. 1). At decision point L4 three persons decided wrong with a metric instruction ("Turn veer right and walk straight for 66 meters"). Since all participants with the landmark instruction ("Turn veer right and walk in the direction of the street and the residential building") decided correctly, we assume that the additional anchoring of actions with landmarks was beneficial for correctly interpreting the turn concept, whereas distance information turned out to be not enough.

Decision point L4a was not identified as decision point by participants of the previous experiment. In this experiment, however, an approximately equivalent number of participants of each test group considered L4a as decision point and decided wrong due to a missing instruction (four persons from the metric group and five persons from the landmark group). These participants followed the direction of the previous route choice whereas nine participants followed the right path.

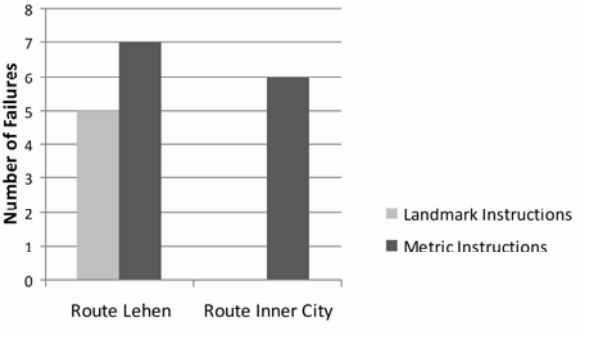


Fig. 4. Number of decision failures along both routes

Beside wrong decisions at decision points L4 and L4a, two participants misinterpreted landmark instructions at decision points L13 and L15. These decisions are a result of misinterpreting the linguistic concepts "Litfasssäule" (eng. advertising column) and "Wohnstrasse" (eng. residential street). In both cases participants did not understand the concept obviously due to linguistic utterances coming from different cultural backgrounds (despite all participants were German native speakers). One solution to this problem could be to further elaborate on linguistic concepts by taking the theory of basic level concepts [30] into account.

Concerning the Inner City Route, wrong decisions occurred only at two decision points. At decision point A13 three persons misinterpreted the instruction from the metric instruction set ("Walk straight for 42 meters"). Again giving additional information by anchoring actions with landmarks helped participants from the landmark group to stay on track ("Walk straight, pass the bus stop and walk in the direction of the fortress"). With the same arguments failures at decision point A25 can be explained. Again three participants getting metric instructions decided for the wrong choice ("Turn veer right and walk 37 meters"), whereas all participants with the landmark-enhanced instruction ("Turn veer right and walk to the statue") decided for the right choice. Concluding from the results, beneficial effects from anchoring actions with landmark can be proven. Results also demonstrate that landmark-based instructions are more beneficial along the Inner City Route compared to the Lehen Route. However, it is also worth to mention, that beside some beneficial effects, most of the instructions (99.98%) were correctly interpreted with either metric- or landmark-enhanced information. Thus we conclude that voice-based navigation performance in terms of error rate is mainly determined by two factors: (1) Accurate turn direction concepts based on wayfinding choremes and (2) position-accurate triggering of voice commands. Additional information has only a minor impact on error rate.

3.4 Qualitative and Quantitative Responses to the Different Instruction Sets

After the experiment participants were asked to indicate their confidence with the two different instruction sets on a 5-point scale. Results are shown in Fig. 5.

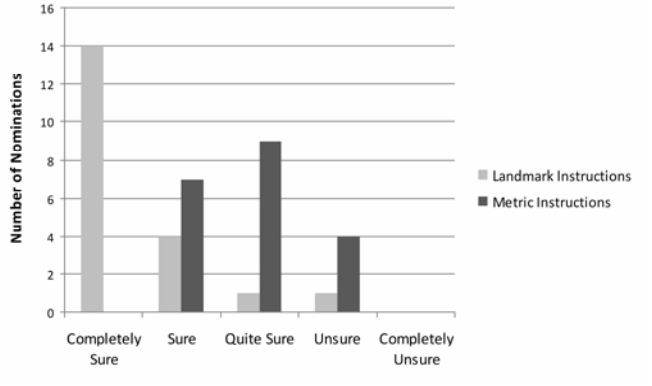


Fig. 5. Confidence rating with the different instruction sets

18 test persons indicated that they were completely sure or sure to be on the right way with the landmark-enhanced instruction set. No participant felt extremely sure with metric instructions, but 16 felt sure or quite sure. Four persons felt unsure with metric instructions but only one person felt unsure with landmark instructions. No person felt completely unsure with either metric or landmark instructions. Considering these results, hypothesis three can be accepted. Participants felt more confident with the landmark instruction set compared to the metric instruction set.

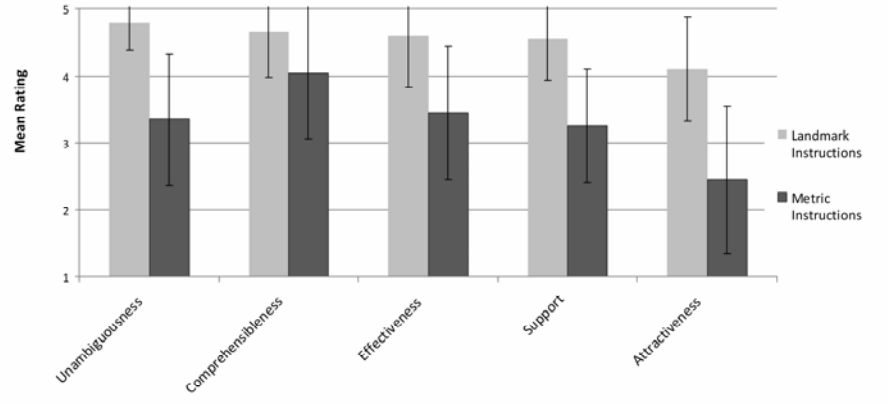


Fig. 6. Mean ratings for different aspects of the verbal instruction sets. 1 indicates a strongly negative and 5 a strongly positive response. Error bars show standard deviations.

In addition to asking participants about their confidence, different aspects of the metric and landmark instruction sets were tested. Fig. 6 shows a clear preference for the landmark instruction set. Especially the aspects *unambiguousness* and *comprehensibleness* were rated strongly positive. In contrast, participants rated the *attractiveness* of metric instructions less positively.

Qualitative questions about personal impressions completed the experiment. Asking about their strategy for navigating a route segment to the next decision point, test persons said that they immediately learned to trust the system, that every direction change will be accurately indicated by a voice instruction. Relying solely upon metric instructions some participants started to count steps in order to estimate segment lengths. Additional information anchoring actions to landmarks increased confidence. Frequency of instructions as well as instruction length was in general evaluated positively. Along path segments A5 – A7 (Inner City Route) and L15 – L17 (Lehen Route), almost all participants complained about too many instructions. These were segments were the direction as well as the reference landmark did not change and thus voice instructions could be generalized. One person suggested including street names as spatial references.

Concerning the turn direction concepts the main directions could be successfully translated into verbal concepts ("straight", "left" and "right"). Participants indicated problems with derived direction concepts like "veer right" or "veer left". Especially the German terms "halb rechts" and "halb links" were confusing for participants. Klippel et al. [31] note that different linguistic concepts of one choreme may be used. A more understandable linguistic expression for "veer left/right" in German language could be "schräg links/rechts". Carefully addressing such linguistic utterances will be crucial for the success of voice-only pedestrian navigation.

4 Discussion

In this paper we investigated the effectiveness of voice-only GPS-based navigation support with two different kinds of instruction sets.

Both instruction sets were composed with cognitively ergonomic turn direction concepts following the wayfinding choreme theory introduced by Klippel et al. [16]. The experiment reveals that adequately used cognitively ergonomic turn direction concepts are the most crucial ingredients of good voice instructions. The value of wayfinding choremes on basis of the revised sector model [27] could be confirmed with a high percentage (99.98%) of right decisions along both test routes. However, the experiments also revealed the importance of accuracy of turn directions with voice-only instructions. If the used concepts are not unambiguously describing route choices, the risk of wrong decision is high (maybe higher as with map support since the overall picture of the route is missing). Problems with turn direction concepts mainly arise from wrongly calculated turn angles as well as misinterpretations due to linguistic utterances. If one linguistic concept is in widespread use by Austrians, this does not mean that Germans make use of the same linguistic concept. However, if mental conceptualizations are matching, different empirically grounded linguistic representations will likely solve the problem.

Measuring walking times as well as decision errors for both instruction sets revealed, that in the context of voice-only GPS-based navigation support highly accurate turn instruction most likely contribute to staying on track and to complete a route in standard walking speed (1.51 m/s). We did not find measurable effects on walking speed coming from different instruction sets, but we did find some minor influences on decision errors. Whereas additional metric information has only minor influence on error rate, landmark-enhanced information has measurable effects.

The experiments revealed that accurate turn directions and timely triggering of instructions have the highest impact on the effectiveness of navigation support. Anchoring actions to visible landmarks helped participants to increase the confidence of selecting the right choice. Especially in decision situations where turn directions are slightly ambiguous, adding landmark-anchored actions showed a positive effect in terms of a decreased error rate. Confirming the importance of landmarks for pedestrian navigation is in accordance with many previous findings (e.g. [24]). However, previous studies did not investigate the effects of landmarks in the context of GPS-based voice-only guidance. The main difference between location-triggered voice instructions and route descriptions is that landmarks are not used to identify decision points (as decision points are automatically determined by GPS positioning), but for identifying route choices as well as path segments in ambiguous decision situations. This slightly different use of landmarks has been positively evaluated in the experiment.

Additionally, the selection of common-sense linguistic concepts for denoting landmarks has been revealed as key issue for successful voice guidance. As findings reveal, wrong decision could be a result of misinterpreting voice instructions due to unknown linguistic concepts. User-generated and thus empirically founded linguistic concepts from our previous experiment have proven to be a good approach to identify candidates for common-sense linguistic concepts. In this context, using folksonomies [32] for denoting landmark categories could bring us the next step up the ladder towards cognitively ergonomic instructions.

Whereas landmark-anchored actions proved to be a useful enhancement of turn directions for all participants, lengths of route segments validated positively only by few of them. Most of the participants trusted in timely voice commands and did not interpret meters. In some cases metric information was interpreted but estimated

wrong (nevertheless participants performed equally due to timely commands), some other participants used the metric information for getting confidence on longer route segments.

Concerning the overall effectiveness of voice-only guidance our results are contrary to findings reported in previous studies, especially the study by Ishikawa et al. [4]. In our experiment, using voice-enabled GPS-based navigation did not lead to longer walking times, except route parts including waiting times at red lights. Also the error rate was surprisingly low. Switching the instruction set from metric to landmark instructions or vice versa during navigation did not have a measurable effect on navigation performance (Ishikawa et al. report on effects in the context of switching reference frames [13]). Our results confirm the assumption stated by Streeter et al. [14] that carefully generated voice instructions show a positive effect on navigation performance, not only in the context of car navigation, but also in the context of pedestrian navigation.

In future work a number of questions have to be tackled. We did not compare our approach with other types of GPS-based navigation support like electronic maps, augmented reality or tactile interfaces. Furthermore, in our experiments we decided to simulate GPS positions in order to guarantee accurate voice commands. An open question is how uncertainties of the GPS signal will influence navigation performance and user confidence. We also left open the question how good voice instructions could be generated automatically out of existing navigational data or out of user-generated content [32] like the OpenStreetMap[2] database. Maybe new data models as well as user-generation processes are necessary in order to collect adequate datasets which can be leveraged for voice-only navigation. However, with our field experiment we contribute the knowledge, that with cognitively ergonomic instructions, an effective voice-only GPS-based pedestrian guidance at minimal error rate and standard walking times is feasible.

References

1. Arikawa, M., Konomi, S., Ohnishi, K.: Navitime: Supporting pedestrian navigation in the real world. IEEE Pervasive Computing 6(3), 21–29 (2007)
2. Baus, J., Cheverst, K., Kray, C.: A survey of map-based mobile guides. In: Meng, L., Zipf, A., Winter, S. (eds.) Map-based mobile services - Theories, Methods, and Implementations, pp. 193–209. Springer, Heidelberg (2005)
3. Knauff, M., Meilinger, T.: Ask for Directions or Use a Map: A Field Experiment on Spatial Orientation and Wayfinding in an urban Environment. Journal of Spatial Science 53(2), 13–23 (2008)
4. Ishikawa, T., Fujiwara, H., Imai, O., Okabe, A.: Wayfinding with a GPS-based mobile navigation system: A comparison with maps and direct experience. Journal of Environmental Psychology 28, 74–82 (2008)
5. Dillemuth, J.: Map Size Matters: Difficulties of Small-Display Map Use. In: Proc. of the 4th Int. Symposium on LBS Services & TeleCartography, Hongkong, pp. 181–191 (2007)
6. Chittaro, L., Burigat, S.: Augmenting Audio Messages with Visual Directions in Mobile Guides: an Evaluation of Three Approaches. In: Proc. of the 7th Int. Conference on Human Computer Interaction with Mobile Devices & Services, Salzburg, pp. 107–114 (2005)

[2] http://www.openstreetmap.org

7. Goodman, J., Brewster, S., Gray, P.: How can we best use landmarks to support older people in navigation? Behaviour & Information Technology 1, 3–20 (2005)
8. Ishikawa, T., Yamazaki, T.: Showing Where To Go by Maps or Pictures: An Empirical Case Study at Subway Exits. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 330–341. Springer, Heidelberg (2009)
9. Coors, V., Elting, C., Kray, C., Laakso, K.: Presenting route instructions on mobile devices: From textual directions to 3D visualization. In: Dykes, J., MacEachren, A.M., Kraak, M.-J. (eds.) Exploring Geovisualization, pp. 529–550. Elsevier, Amsterdam (2005)
10. Reitmayr, G., Schmalstieg, D.: Collaborative augmented reality for outdoor navigation and information browsing. In: Proc. of Symposium Location Based Services and TeleCartography, Geowiss. Mitteilungen, vol. 66, pp. 53–62. Eigenverlag, Austria (2004)
11. Heuten, W., Henze, N., Boll, S., Pielot, M.: Tactile Wayfinder: A Non-Visual Support System for Wayfinding. In: Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges, pp. 172–181. ACM, New York (2008)
12. Holland, S., Morse, D., Gedenryd, H.: AudioGPS: Spatial Audio Navigation with a Minimal Attention Interface. Personal and Ubiqu. Computing 6(4), 253–259 (2002)
13. Ishikawa, T., Kiyomoto, M.: Turn to the Left or to the West: Verbal Navigational Directions in Relative and Absolute Frames of Reference. In: Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 119–132. Springer, Heidelberg (2008)
14. Streeter, L.A., Vitello, D., Wonsiewicz, S.A.: How to tell people where to go: Comparing navigational aids. International Journal of Man-Machine Studies 22, 549–562 (1985)
15. Klippel, A., Richter, K.-F., Hansen, S.: Cognitively ergonomic route directions. In: Karimi, H.A. (ed.) Handbook of Research on Geoinformatics, pp. 230–238. Idea Group Inc., Pittsburgh (2009)
16. Klippel, A., Tappe, T., Kulik, L., Lee, P.: Wayfinding choremes - A language for modeling conceptual route knowledge. J. of Visual Languages and Comp. 16(4), 311–329 (2005)
17. Denis, M.: The description of routes: A cognitive approach to the production of spatial discourse. Current Psychology of Cognition (16), 409–458 (1997)
18. Denis, M., Michon, P.-E., Tom, A.: Assisting Pedestrian Wayfinding in Urban Settings: Why References to Landmarks are Crucial in Direction Giving. In: Allen, G.L. (ed.) Applied spatial cognition: from research to cognitive technology, pp. 25–51. Lawrence Erlbaum Associates, New Jersey (2006)
19. Lovelace, K.L., Hegarty, M., Montello, D.R.: Elements of Good Route Directions in Familiar and Unfamiliar Environments. In: Freksa, C., Mark, D. (eds.) Spatial information theory: Cognitive and computational foundations of geographic information science, pp. 65–82. Springer, Berlin (1999)
20. Millonig, A., Schechtner, K.: Developing Landmark-Based Pedestrian Navigation Systems. IEEE Transactions on Intelligent Transportation Systems 8, 43–49 (2007)
21. Lynch, K.: The Image of the City. MIT Press, Cambridge (1960)
22. Sorrows, M.E., Hirtle, S.C.: The Nature of Landmarks for Real and Electronic Spaces. In: Freksa, C., Mark, D.M. (eds.) COSIT 1999. LNCS, vol. 1661, pp. 37–50. Springer, Heidelberg (1999)
23. Raubal, M., Winter, S.: Enriching Wayfinding Instructions with Local Landmarks. In: Egenhofer, M.J., Mark, D.M. (eds.) GIScience 2002. LNCS, vol. 2478, pp. 243–259. Springer, Heidelberg (2002)
24. Ross, T., May, A., Thompson, S.: The use of landmarks in pedestrian navigation instructions and the effects of context. In: Proc. of the 6th Int. Symp. of Human Computer Interaction with Mobile Devices and Services, pp. 300–304. Springer, Berlin (2004)

25. Knoblauch, R.L., Pietrucha, M.T., Nitzburg, M.: Field studies of pedestrian walking speed and start-up time. Transportation Research Board Records No. 1538 (1996)
26. Rehrl, K., Leitinger, S., Gartner, G., Gartner, G., Ortag, F.: An analysis of direction and motion concepts in verbal descriptions of route choices. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 471–488. Springer, Heidelberg (2009)
27. Klippel, A., Dewey, C., Knauff, M., Richter, K.-F., Montello, D., Freksa, C., et al.: Direction Concepts in Wayfinding Assistance Systems. In: Baus, J., Kray, C., Porzel, R. (eds.) Workshop on AI in Mobile Systems (AIMS 2004), Saarbrücken, pp. 1–8 (2004)
28. Ishikawa, T., Yamazaki, T.: Showing Where To Go by Maps or Pictures: An Empirical Case Study at Subway Exits. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 330–341. Springer, Heidelberg (2009)
29. Rogers, Y., Connelly, K.H., Tedesco, L., Hazlewood, W., Kurtz, A., Hall, R.E., Hursey, J., Toscos, T.: Why It's Worth the Hassle: The Value of In-Situ Studies When Designing Ubicomp. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 336–353. Springer, Heidelberg (2007)
30. Rosch, E., Loyd, B.B.: Cognition and Categorization. Lawrence Erlbaum Associates, Hillsdale (1978)
31. Klippel, A., Montello, D.R.: Linguistic and Nonlinguistic Turn Direction Concepts. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds.) COSIT 2007. LNCS, vol. 4736, pp. 354–372. Springer, Heidelberg (2007)
32. Quintarelli, E.: Folksonomies: power to the people. Paper presented at the ISKO Italy-UniMIB meeting (2005), http://www.iskoi.org/doc/folksonomies.htm
33. Holone, H., Misund, G., Holmstedt, H.: Users are Doing It for Themselves: Pedestrian Navigation With User Generated Content. In: Next Generation Mobile Applications, Services and Technologies, pp. 91–99. IEEE Computer Society, Los Alamitos (2007)

A Mismatch Description Language for Conceptual Schema Mapping and Its Cartographic Representation

Thorsten Reitz

Fraunhofer Institute for Computer Graphics Research,
Fraunhoferstr. 5, 64283 Darmstadt, Germany
thorsten.reitz@igd.fraunhofer.de

Abstract. Geospatial data offered by distributed services are often modeled with different conceptual schemas although they cover the same thematic area. To ensure interoperability of geospatial data, the existing heterogeneous conceptual schemas can be mapped to a common conceptual schema. However, the underlying formalized schema mappings are difficult to create, difficult to re-use and often contain mismatches of abstraction level, of scope difference, domain semantics and value semantics of the mapped entities. We have developed a novel approach to document and communicate such mismatches in the form of a Mismatch Description Language (MDL). This MDL can be transformed into various textual and cartographic representations to support users in communicating and understanding mismatches, and to assess the reusability of a mapping.

Keywords: Conceptual Schema Mapping, Mismatch Description, Mismatch Identification, Data Integration, Transformed Data Quality.

1 Introduction and Motivation

Conceptual Schema Mapping (CSM) is the process of defining the relationships between classes, properties and relations of two heterogeneous conceptual schemas and is considered as an important means of achieving data integration [1, 2]. Schema mapping is used to validate the logical consistency of the mapping with respect to the mapped schemas and to perform the actual instance transformation. A mapping for this purpose is unidirectional, from a source to a target schema.

A typical integration scenario can involve geographic data sets from different countries, each of which with its own conceptual schema, terminology and code lists [3]. As an example, consider a user who needs to have an integrated hydrographic network of a region where there are several different data providers who are responsible for delivering hydrographical data, each using a different conceptual schema. Mapping all those schemas to a common, harmonized one will allow the user to transform the data provided by these organizations and to use it in his context.

However, creating such a conceptual schema mapping is a complex task, and it often involves inaccuracy because the concepts mapped have originally been created on the basis of different universes of discourse, with different application cases in mind and using different conceptual schema languages. These and other factors lead to conceptual schema mappings that are based on assumptions and known compromises.

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 204–218, 2010.

If a specific relation is declared despite differences in scope and definition of the mapped concepts, this mapping is called a *mismatch*. Mismatches are often not an issue for a single application context, but they limit the re-usability of the mappings for any other context than the original one for which it was created. The goal of the presented approach is to increase the re-usability of conceptual schema mappings as well as of the transformed datasets.

Early approaches for expressing schema mappings had a relatively simple structure and were often limited to basic types of relations between concepts, such as subsumption and equivalence [4, 5]. Equivalence, for example, was often qualified by employing a single numeric value (e.g. in the range of 0...1, with 0 being no equivalence and 1 being complete equivalence). Such one-dimensional qualification values cannot communicate mismatches in the mapping. They have their use primarily for expressing result confidence in automated alignment approaches. Further, these mapping languages do not contain enough information to facilitate instance transformations.

More current approaches, such as the Ontology Mapping Language (OML) developed by Scharffe [6] or Agarwal's approach for reconciling ontologies [7], make schema mappings more powerful and allow the definition of mappings in such a way that fewer mismatches are introduced. An additional advantage of the OML is that it can be used with a wide range of conceptual schema languages, such as OWL, UML or GML Application Schemas. OML is also expressive enough to be used for query rewriting and instance transformation.

However, these approaches do not make an effort to include *irreconcilable mismatches* – mismatches that are taken into account consciously by an expert creating a mapping – into the mapping. The possibility to include mismatch information would have several advantages, e.g. supporting quality assurance and communicating limitations of the mapping to others. The information can also be used for inference purposes, and it can be used to document mismatches in transformed geographic datasets as part of that dataset's lineage.

The core requirements we identified for such a Mismatch Description Language (MDL) and the model behind it are therefore:

- To allow a person performing conceptual schema mapping *to document limitations of the mapping* and mismatches that are knowingly taken into account;
- To *communicate mismatches* both to persons performing schema mapping and to persons who want to use data that was transformed using a schema mapping;
- To provide the possibility to *represent the mismatch descriptions visually* or in understandable text form through the application of a set of (transformation) rules.
- When transforming a geospatial data set using the schema mapping with included *mismatch information*, this information *has to become metadata of the resulting data set*.
- It should be possible to *use the information in mismatch descriptions for inferencing* in established reasoning environments.

We therefore propose to complement newer mapping languages like the OML with an approach to document mismatches that fulfills the requirements listed above, with the goal of increasing the transparency of decisions made during the mapping process and of creating higher-quality conceptual schema mappings. In this paper we provide an analysis of mismatches and their consequences in geospatial data integration and describe the specification of a Mismatch Description Language. Furthermore, we show how the Mismatch Description Language can be used to communicate mismatches and provide initial evaluation results.

The sections of this paper are organized as follows: First, an analysis of mismatches that do occur in the mapping of conceptual schemas of geospatial datasets is conducted. Then, the possible consequences that mismatches can have in terms of the fitness-for purpose of a transformed data set are outlined. On this basis, the Mismatch Description Language is modeled. In a next step, means for using the MDL to satisfy the requirements described in this section are laid out. The paper concludes with a summary and outlook section.

2 Mismatch Types and Occurrences in the Geospatial Domain

In our understanding, mismatches result from divergences in characteristics of the mapped schema entities. We group such characteristics of schema entities as follows:

- *Formalised characteristics:* Characteristics that have been modeled formally in the conceptual schema, including the following:
    - Relations to other schema entities, including generalization, aggregation, spatial and non-spatial constraints;
    - Data type used, including constraints on allowable values;
    - Cardinality on relations and values;
- *Defined characteristics:* Characteristics that have not been formalized, but agreed upon, often in written from, such as a Feature Catalogue that contains detailed classification rules for land cover and use classifications. Thus, differences in a defined characteristic can lead to instances being assigned to different classes, even when their formalized properties are identical.
- *Interpreted characteristics:* Characteristics that are neither modeled formally nor defined, but rather stem from the interpretation of the defined characteristics by a person. The extent of such characteristics depends on the level of ambiguity present in the defined characteristics.

To identify types of mismatches and their properties, we have collected mismatches described in literature by Agarwal [8]; Visser et al. [9]; Scharffe [6]; Predoiu et al. [10] and Klein [11] and have evaluated their appearance in six geospatial integration scenarios. Consequently, the catalogue as represented here emphasizes the core mismatches that we identified as being relevant for those typical geospatial schema integration scenarios. It omits mismatches related to terminology usage, e.g. usage of homonyms (same words, but different concepts depending on context) and mismatches that can usually be fully resolved employing existing approaches.

To illustrate the mismatches and their possible consequences, examples from a hydrographic data integration scenario are used. In the figures adjoining the examples, the source schema elements are always shown with bright background, the target schema elements with a darker background.

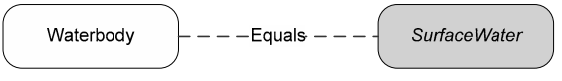
For each mismatch, a title, an explanation why it occurs based on the schema element properties outlined above, a verbal description of the resulting consequences and examples are given.

2.1 Scope Mismatches

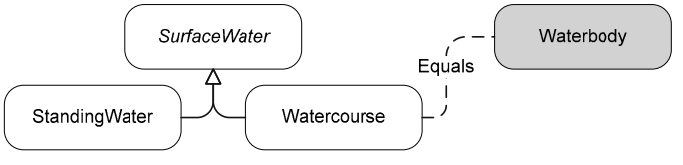
These mismatches are characterized through the fact that the extension of a class is different in the source and target schema.

2.1.1 Subsumption/Abstraction Mismatch

- *Reason:* The extension of a class in the source schema is a subset of the extension of a class in the target schema, or vice versa. These mismatches occur when the source and target schema define classes at a different level of abstraction.
- *Consequence*: When the extension of source schema class is a superset of the extension of the target class, not all instances of the source class can be translated to the target class without impacting completeness (excess). When the extension of the target schema class is a superset of the extension of the mapped source schema class, the completeness of the resulting data set is lowered.
- *Examples*:



A class `Waterbody` is mapped as equal to an abstract class `SurfaceWater`. However, the `Waterbody` class also includes subsurface waterbodies and therefore represents a superset of the instances of `SurfaceWater`.

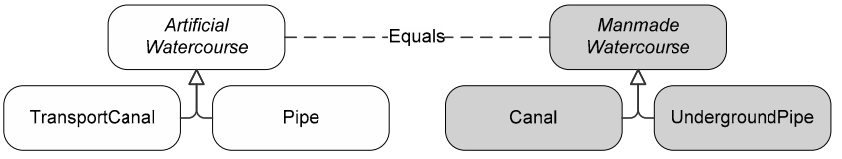


A class `Watercourse` is mapped as equal to the class `Waterbody`. However, since `Waterbody` should encompass `Watercourses`, `StandingWaters` and `SubsurfaceWaters`, this mapping includes a mismatch with a consequence on the completeness of the resulting data sets – a user might expect to get all types of water bodies, but will get only watercourses.

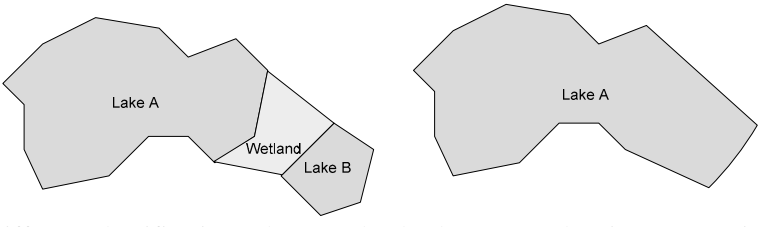
2.1.2 Overlapping Scope Mismatch

- *Reason:* The extension of the class in the source schema and the extension of the class in the target schema are not identical, but have a non-empty intersection.
- *Consequence:* A part of the extension of the class in the source schema cannot be transferred to the classes in the target schema and will get lost, possibly impacting completeness and classification correctness.

- *Examples:*



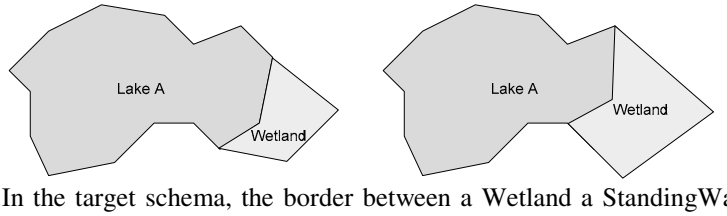
The classes `ArtificalWatercourse` and `ManmadeWatercourse` are mapped as equal. However, when looking at the subtypes, it becomes clear that they only share the instances falling within the more confined subclasses `UndergroundPipe` and `TransportCanal`. Pipes on the ground or irrigation canals are not part of the shared instance set.



Different classification rules can also lead to an overlapping scope mismatch. In the source dataset excerpt, there are `Lakes` and `Wetlands` as separate features, whereas in the target schemas separate wetlands are only shown when they are bigger than a certain threshold. The same applies to Lake B, which is below a threshold and is therefore included in the Lake A instance. This variant of the *Overlapping Scope Mismatch* is connected to the *Categorization Mismatch*.

2.1.3 Categorization Mismatch

- *Reason*: Different rules are applied in the classification, e.g. space is partitioned differently due to different classification rules, but there is the same number of instances in source and target.
- *Consequence*: While the basic set of instances is equal, the properties that are the basis of the categorization have different values, and therefore their precision is degraded.
- *Example:*

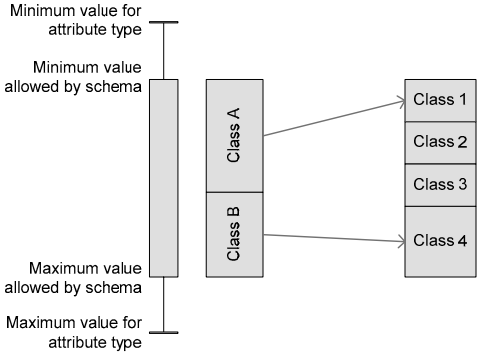


In the target schema, the border between a Wetland a StandingWater is defined differently than in the source Schema, and leading to a different geometric representation of the instances.

2.1.4 Aggregation-Level Mismatch

- *Reason*: Classification of allowable values for a schema entity on the formalization or definition level has a different granularity.

- *Consequence*: When the source classification is coarser than the target classification: classification precision of data is less accurate than indicated by the classification in the target schema. When the target classification is coarser than source classification, classification precision is degraded since classes are aggregated, especially if the value classes don't have shared borders.
- *Example*:



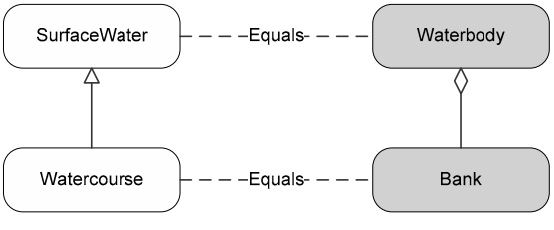
For a given attribute of data type *Short*, the minimum value is -32,768, whereas the maximum value is 32,767. The minimum value allowed by the schema in this example is 0, the maximum value 10,000. The border between `Class A` and `Class B` in the source schema is put at a value of 20. When mapping `Class A` and `B` to the `Classes 1…4` (which split the value range in 0…5, 5…10, 10…25 and 25…) in the target Schema, a decision has to be made how to reclassify instances. In this example, this will lead to a high classification precision error, since all instances that had been in `Class A` (0…20) are now put into `Class 1` (0...5), and a lower precision error for the mapping of `Class B` (20...) to `Class 4` (25…).

2.2 Relation Mismatches

These are mismatches that occur because relations between classes are modeled differently in the source and the target schema.

2.2.1 Structure Mismatch

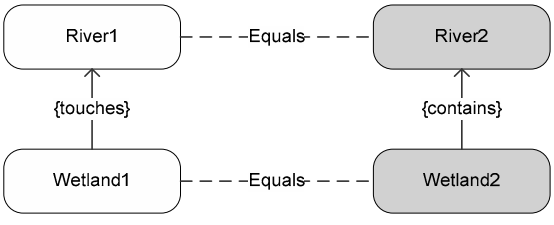
- *Reason*: The structural relation between two classes As and Bs in the source schema is different (or has a different meaning) than the relation between two corresponding classes At and Bt in the target schema.
- *Consequence*: The mapping and the transformed data is not logically consistent, resulting in a lowered conceptual consistency.
- *Example*:



In the source schema, `Watercourse` is a subtype of `SurfaceWater`, and in the target schema, `Waterbody` aggregates Bank. However, since `SurfaceWater` and `Waterbody` and `Watercourse` and `Bank` have been mapped as Equal, the mapping is logically inconsistent.

2.2.2 Constraint Mismatch

- *Reason*: Source schema is under-constrained in comparison to target schema, or uses different constraints.
- *Consequence*: Constraints defined on the target schema can be violated, thus, conceptual consistency within its scope is not ensured. Also, if the constraint refers to a spatial property the geometrical consistency and the topological consistency are impacted.
- *Examples*: Attribute in target schema that must not be null, but corresponding attribute in source does not have such a constraint.



Another example based on spatial constraints: In the source schema, the relation between `Wetland1` and `River1` is `Wetland1` touches `River1`, whereas in the target schema, the relation is `Wetland2` contains `River2`.

2.2.3 Attribute Type and Encoding Mismatches

- *Reason*: The value range of an attribute varies due to different units of measurement or data types
- *Consequence*: The actual accuracy of the property can be reduced.
- *Example*: In the source schema, an `int` value is used to express meters, whereas in the target schema, a `double` is used, which allows to express millimeter accuracy. In another example, consider a geometric property modeled as a simple polygon in the source and as a complex `Surface` (with holes) in the target.

2.3 The Impact of a Mismatch: Consequences

The impact that a certain mismatch has largely depends on the context in which data transformed based on a given mapping is going to be used. As an example, consider an *Aggregation-level Mismatch* on a width classification property of a River, where a property value classification in one schema is much coarser (having only two classes, one for rivers under 12m width and one for rivers over 12m width) than the corresponding classification (where there are five classes for width) in the mapped schema. This difference is only important if for the context in which the transformed data is going to be used, a high accuracy of the classification is required.

To allow the identification of when a certain consequence is of importance, we define both the context and the consequence on the basis of *quality elements* as they are also used in metadata for geospatial data [12]. These quality elements provide a

model to describe the quality of a given data set and include generic properties like *Completeness* and *Conceptual Consistency*, but also elements that are specific to geographical data such as *Positional Accuracy* and *Topological Consistency*. In a consequence, these quality elements are used to denote a change in one of the properties. As an example, consider a case of a *Categorisation Mismatch* where the classification rules that define the border of a waterbody vary in the mapped schemas. This will lead to a reduced *Positional Accuracy* for any transformed data, since the source data didn't use the same rules as those which are defined for the target schema.

3 Linking Mismatches and Consequences – Introducing the MDL

In addition to the mismatches described in section 2, there are multiple other types of mismatches that can be circumvented using current mapping languages, such as OML. This language was specifically developed to allow the definition of mappings that are expressive enough so that some types of mismatches, such as the *Subsumption Mismatch* and some other mismatches, e.g. the *Attribute Assignment Mismatch* and the *Element Partitioning Mismatch* can be resolved fully. However, some mismatches, such as the *Categorization Mismatch* and the *Aggregation-level Mismatch* cannot be resolved completely using the mapping constructs. Furthermore, there are mismatches that are knowingly taken into account or accidentally inserted by the person defining the conceptual schema mapping.

These mismatch types and the mismatches taken into account knowingly are characterized by aspects that are not part of the formalized schema being mapped, but result of specific non-formalized domain knowledge. It therefore has to be within the scope of the MDL to provide means to capture these complex constraints. In the MDL, each mismatch is defined as a triple (M, R, C): A Mismatch M occurs because of reason R and has consequence(s) C.

1. *Mismatch M*: Identify the mismatch type that occurs.
2. *Reason R*: Document why a certain mismatch occurs in relation to characteristics of the mapped schema entities, such as their relations to other schema entities, types and constraints of values or the set of instances assigned to the entity.
3. *Consequence C*: Support the automatic analysis of the quality of the mapping and the ensuing transformation.

This relatively basic structure can be used to extend most mapping languages. Because of its expressivity and its loose coupling to conceptual schema languages, we chose to extend the Ontology Mapping Language [6]. This RDF-based language has an `Alignment` element that contains information on the two mapped conceptual schemas, such as their namespace and the formalism the schemas have been expressed with. The core unit of an alignment is a `Cell`, which maps two `Entities`. The Cell is the extension point where the MDL is inserted: Mismatches occur per `Cell`, and have their `Reason` in the composition and selected relation of the `Entities` belonging to that `Cell`. Consequences are expressed in relation to quality elements, and the `Schema` elements

of the OML as `Context`. Such a context can be used to denote that the documented mismatch will matter (or not) within a specific conceptual schema. This feature can be used by the expert creating the mapping to document the impact of mismatches. The UML class diagram in Figure 1 shows this structure in an object-oriented view.

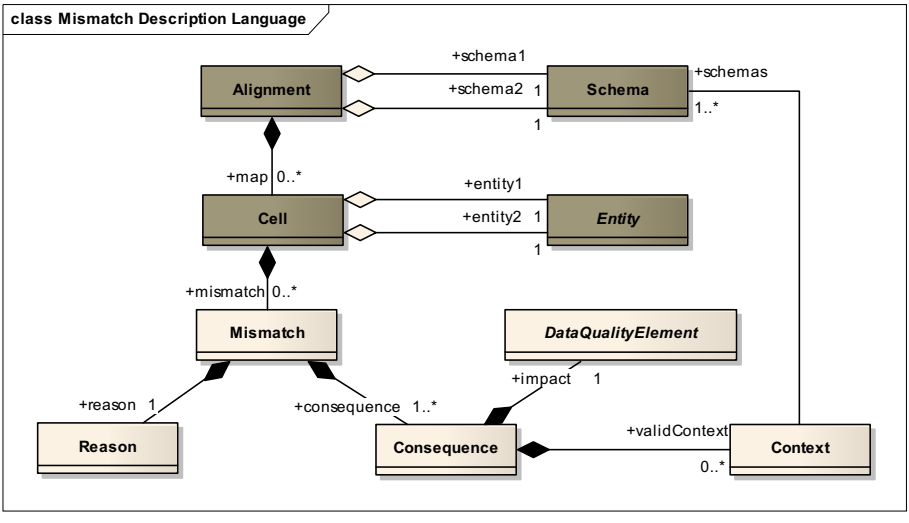


Fig. 1. The main structure of the MDL. OML elements are shown with a dark background.

To maintain format consistency with OML, XML is also used for the encoding of mismatches. Listing 1 gives an example of an encoded mismatch.

Listing 1. An example for an encoded mismatch, in the scope of an attribute mapping cell which relates a *names* and a *geographicalNames* property

```
<mismatch type="ConstraintMismatch">
  <reason>
     <source>cardinality:*</source>
     <target>cardinality:1..*</target>
     <description>
       Entity1 (names) may contain any number of elements, but
       Entity2 (geographicalNames) must contain 1 or more elements.
     </description>
  </reason>
  <consequence>
    <conceptualConsistency>
      <impact><validationResult>false</validationResult></impact>
      <description>
        Cardinality constraint on Entity2 (1..*) can be violated.
      </description>
    </conceptualConsistency>
  </…>
```

4 Inferencing on the Basis of Mismatches

It would be idealistic to expect a user engaged in a schema mapping process to fully provide the information on mismatches as described above. Therefore, we have evaluated possibilities to automatically identify likely mismatches and to instantiate parts of the mismatch, such as the *Reason* part. Using rule-based inference with a knowledge base built of the mapped schemas (including information on the classified instances, if available), the mapping and the mismatch descriptions offers several possibilities:

- The identification of likely mismatches and subsequent automated instantiation of MDL elements, especially of likely consequences, to support the interactive mapping process;
- Validate the logical and semantic consistency of the mapping in relation to the mapped schemas;
- Analysis of whether a mapping can be used in a different context than the one for which it was originally created.

The rule-based approach to inference was chosen because it can be extended well, also with rather complex rules, and because it can be extended to work on the basis of instances, e.g. for validating transformation consistency with less expressive schemas. However, there are several prerequisites that need to be fulfilled for using inference towards these goals:

- *The mapped schemas have to be expressive enough*. This can be a problem when formalisms for expressing the schema do not use a rich relation or constraint model or when constraints were simply not made explicit.
- Since reasoning over schemas expressed in heterogeneous schema languages is usually not possible, *the mapping, the source and the target schema have to be "lifted" to a homogeneous, expressive conceptual schema language*.
- *Indicators for mismatches have to be formalized* in a form matching the homogeneous conceptual schema language and digestible by the inference engine to be used. Often, there is more than one possible reason type for a given mismatch (and many rules to find out which consequences are likely to result), so providing a full library of formal mismatch descriptions will be a continuous task.
- *Non-taxonomic relations that are to be used in inference have to be defined formally* as well, such as the spatial relationships used to express constraints between classes [13].

Even when these prerequisites are fulfilled, there would still be some mismatches that are hard to identify. Without a schema declaring relations between the elements of the mapped schemas, such as a top-level ontology, it is possible for a user to declare two classes A and B equivalent that are, in fact, disjoint, without any inferencing noticing this mismatch. In some cases, the mapping itself and the constraints defined within one of the mapped schemas can be used to infer an incompatible scope mismatch. The identification of mismatches is the first step – the logical next step is inferring the consequences that result out of an identified mismatch. For each mismatch and each

consequence, there can be multiple rules that, when fulfilled, give indication that a given consequence is relevant.

Listing 2. A simple example for a mismatch identification rule that is evaluated whenever a new cell relating two entities (attributes) is added to a mapping

```
when
   cell.entity1.cardinality != cell.entity2.cardinality
then
   addMismatch(new ConstraintMismatch(Type.Cardinality))
```

Using rules such as the one in Listing 2, indications can be given to users what parts of a mapping they should analyze in detail. Users can then either complete the information on the mismatch, can flag it as irreconcilable (and therefore, consciously taken into account) or can flag it as irrelevant. Table 1 summarizes indications which mismatches can be identified by inference providing that the prerequisites described above are met.

Table 1. Mismatches and their identification by inference

	Identification of Mismatch by inference based on...		
Mismatch	**intension + mapping**	**extension**	**Comments**
Subsumption/Abstraction	Yes	Yes	Possible if a filtering condition on a mapping is defined
Overlapping Scope	Yes	Yes	Possible if a filtering condition on a mapping is defined
Incompatible Scope	Partially	No	Not possible if the scope was incompatible between mapped elements of different schemas
Categorisation	No	Yes	This type of mismatch can be identified using reference data which uses the target schema
Aggregation-level	Yes	Yes	A mismatch can generally be assumed to occur when classification mapping is taking place.
Structure	Yes	No	Results depend on the inference approach
Constraint	Yes	No	Possible if a filtering condition on a mapping is defined
Attribute Type and Encoding	Yes	No	Can be identified automatically when types of bound elements are not fully compatible

The final step is the determination whether a consequence is of importance for the context in which the mapping or data transformed is being used. This can be done by comparing the quality requirements expressed for the target context with the consequences using a fixed set of rules. However, to do so, the original data set must also have had quality metadata, since consequences can only express changes to quality.

5 Representations of Mismatches

To communicate the mismatches formalized with the MDL, there are several possibilities to translate the formal description into readily understandable representations. The selection of a representation depends on the availability of transformed instance data, reference data and the type of this data.

Textual Representations. A textual representation is simple to generate from an MDL object and can itself become part of the MDL object. The mismatch in Listing 1 contains automatically generated descriptions of the mismatch reason and the mismatch consequence. Especially for mismatches of lower complexity, generated

textual representations can be very short and comprehensible. These forms can be expressed using different natural languages. Options include:

- *Single-sentence*: Provide the core mismatch information (reason and consequence) in normal text, as a single sentence, with variables and parameters highlighted: *"*names *may contain any number of elements, but* geographicalNames *must contain 1 or more elements, therefore a Cardinality constraint on* geographicalNames *can be violated in transformation."*
- *Tabular*: Provide the information on the currently existing mismatches in tabular form, with one mismatch per row and the properties of the mismatch in columns (or vice versa). This form can e.g. be valuable to quickly order mismatches by any of their properties.
- *Tabular-hierarchical*: This representation is comparable to the tabular form, but columns are themselves hierarchically structured. Such a representation is valuable when information has multiple levels of detail or is organized in a hierarchical way, since it allows first getting an overview and then going into detail where necessary.

Fig. 2. This figure shows two styles of rendering uncertainty introduced with a mismatch in the map. The left one uses width and opacity to display to express a loss of positional accuracy induced by an aggregation-level mismatch, the right one uses circle glyphs to display the same consequence. The relative error induced is used as a control parameter for the visualization.

Cartographic Representations. Integration of geospatial data offers the unique possibility to visually represent mismatches using a map. To be able to do so, instance data in the source schema, and ideally also in the target schema, has to be available. Since consequences effectively describe uncertainty introduced into data, visualization techniques for highlighting uncertainty in geospatial data can be used, for which many usable approaches can be found in the literature [14-16].

Thomson et al. [17] have developed a typology for visualizing uncertainty in geospatially referenced information that is based on the same quality parameters used in this paper to formalize consequences. This typology provides numerous examples for how probabilistic representations can be used for different quality parameters, e.g. using the expected distribution of measurement error to control the concrete visualization.

Specifically in data integration, it is also possible to contrast data in the source schema to data in the target schema. This comparative visualization can be effective in communicating the impact of a mapping when it is interactive, i.e. when it changes with every change to the mapping. Contrasting the transformed data to the original data can help a user to identify and assert mismatches such as the *Categorization Mismatch* and the *Aggregation-level Mismatch*.

Fig. 3. A comparative visualization can also help to assert the amount of uncertainty induced into the data when applying the currently specified mapping

Non-cartographic Visual Representations. Not using cartographic representations can have significant merits, since no dimensions have to be added to an already complex image and therefore, individual properties of the transformed data can be highlighted. Riveiro [18] theoretically evaluated several different types of uncertainty visual representations for their possible merit to generic information fusion.

6 Conclusion and Future Work

The main contribution of this paper lies in the provision of an approach for identifying and documenting mismatches in the course of conceptual schema matching. The discussed approach can be used to support data integration experts in identifying and evaluating mismatches inherent in mappings they create. In such a context, likely mismatches can be identified automatically and be made obvious to the expert carrying out the schema mapping using the MDL. Therefore, unwanted mismatches in subsequent transformations can be avoided or significantly reduced.

For evaluation of the MDL effectiveness in capturing mismatches, their reasons and their consequences, conceptual schema mappings in the form of matching tables from six data integration scenarios were analyzed together with domain experts. The following table presents the outcome of this analysis.

Table 2. Types and Consequences of 91 identified mismatches; (+) indicates that the consequence did occur in at least one of the mismatches of that type, and (++) indicates that it occurred multiple times and in more than 50% of the mismatches of that type.

Mismatch	Positional Accuracy	Completeness – Omission	Completeness – Excess	Conceptual Consistency	Domain Consistency	Format Consistency	Geometrical Consistency	Topological Consistency	Classification Correctness	Qualitative attribute corr.	Quantitative attribute corr.
Subsumption/Abstraction	+	++	++	+					++		
Overlapping Scope	+	++									
Incompatible Scope		++	++								
Categorisation	+			+	+		+	+	++		
Aggregation-level	+			+	+				++	++	++
Structure				++	+					+	
Constraint				++			++	++		+	+
Attribute Assignment		++	+		+						
Attribute Type and Encoding	++				++	+				+	++
Element Partitioning		+	+								
Schema language				+		++					

The OML mappings corresponding to the matching tables were enriched with MDL documentation in an interactive schema mapping application and used to create lineage metadata in a schema translation performed using the mappings. Furthermore, a framework for the transformation of MDL elements into cartographic and textual representations was implemented as part of the same conceptual schema mapping application [19].

In future research, the methods applied for identifying mismatches via inference and especially the rule system used to determine consequences have to be extended and made more stable. Furthermore, especially the non-cartographic visual representations suggested for information fusion have to be thoroughly tested for their efficiency in communicating mismatches in matching geospatial conceptual schemas.

Acknowledgements

I would like to thank Eva Klien and Daniel Fitzner for their support as well as the Landesvermessungsamt of Vorarlberg for providing data to work with. This work was partially funded under the HUMBOLDT Project, EC contract SIP5-CT-2006-030962.

References

1. Mohammadi, H., Rajabifard, A., Williamson, I.P.: Enabling spatial data sharing through multi-source spatial data integration. In: Proceedings of GDSI, Rotterdam, vol. 11 (2009)
2. Zhao, T., Zhang, C., Wei, M., Peng, Z.: Ontology-Based Geospatial Data Query and Integration. Geographic Information Science, 370–392 (2008)
3. Hobona, G., Attardo, C., Laurini, R., Jackson, M., Pla, M., Zorzi, S.D., Breu, A.: Considerations for Harmonising Cross-Border Geospatial Datasets. Presented at the AGILE 2009 Pre-Conference Workshop: Challenges in Geospatial Data Harmonisation, Hannover, Germany (2009)

4. Ehrig, M.: Ontology Alignment: Bridging the Semantic Web Gap. Springer, Berlin (2006)
5. Euzenat, J., Bach, T.L., Barrasa, J., Bouquet, P., Bo, J.D., Shvaiko, P.: State of the art on ontology alignment. Technical Report, INRIA, Innsbruck (2004)
6. Scharffe, F.: Correspondence Patterns Representation. Dissertation, University of Innsbruck (2008)
7. Agarwal, P., Huang, Y., Dimitrova, V.: Formal Approach to Reconciliation of Individual Ontologies for Personalisation of Geospatial Semantic Web. In: Rodríguez, M.A., Cruz, I., Levashkin, S., Egenhofer, M.J. (eds.) GeoS 2005. LNCS, vol. 3799, pp. 195–210. Springer, Heidelberg (2005)
8. Agarwal, P.: Ontological considerations in GIScience. International Journal of Geographical Information Science 19, 501–536 (2005)
9. Visser, P.R., Visser, P.R.S., Jones, D.M., Bench-capon, T.J.M., Shave, M.J.R.: An Analysis of Ontology Mismatches; Heterogeneity Versus Interoperability. Presented at the AAAI 1997 Spring Symposium on Ontological Engineering, Stanford (1997)
10. Predoiu, L., Polleres, A., Martin-Recuerda, F., Feier, C., Mocan, A., Bruijn, J.D., Porto, F., Foxvog, D., Zimmermann, K.: Framework for representing ontology networks with mappings that deal with conflicting and complementary concept definitions. Technical report, DERI, Galway (2004)
11. Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: Proceedings of the Workshop on Ontologies and Information Sharing, IJCAI 2001 (2001)
12. Shi, W., Fisher, P., Goodchild, M.F.: Spatial Data Quality. CRC Press, Boca Raton (2002)
13. Mäs, S.: Reasoning on Spatial Relations between Entity Classes. In: Proceedings of the 5th international conference on Geographic Information Science, pp. 234–248. Springer-Verlag, Park City (2008)
14. Pang, A.: Visualizing uncertainty in geo-spatial data. In: Proceedings of the Workshop on the Intersections between Geospatial Information and Information Technology (2001)
15. Griethe, H., Schumann, H.: The Visualization of Uncertain Data: Methods and Problems. In: Simulation und Visualisierung 2006, pp. 143–156. SCS Publishing House e.V., Magdeburg (2006)
16. MacEachren, A.M., Robinson, A., Gardner, S., Murray, R., Gahegan, M., Hetzler, E.: Visualizing geospatial information uncertainty: What we know and what we need to know. Cartography and Geographic Information Science 32, 139–160 (2005)
17. Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., Pavel, M.: A typology for visualizing uncertainty. In: Visualization and Data Analysis 2005, pp. 146–157. SPIE, San Jose (2005)
18. Riveiro, M.: Evaluation of uncertainty visualization techniques for information fusion. In: Proceedings of the 10th International Conference on Information Fusion, Quebec, Canada, pp. 1–8 (2007)
19. Reitz, T., Kuijper, A.: Applying Instance Visualisation and Conceptual Schema Mapping for Geodata Harmonisation. In: Advances in GIScience, pp. 173–194 (2009)

Detecting Change in Snapshot Sequences

Mingzheng Shi and Stephan Winter

Department of Geomatics, University of Melbourne, Victoria, 3010, Australia
m.shi@pgrad.unimelb.edu.au, winter@unimelb.edu.au

Abstract. Wireless sensor networks are deployed to monitor dynamic geographic phenomena, or objects, over space and time. This paper presents a new spatiotemporal data model for dynamic areal objects in sensor networks. Our model supports for the first time the analysis of change in sequences of snapshots that are captured by different granularity of observations, and our model allows both incremental and non-incremental changes. This paper focuses on detecting qualitative spatial changes, such as merge and split of areal objects. A decentralized algorithm is developed, such that spatial changes can be efficiently detected by in-network aggregation of decentralized datasets.

Keywords: wireless sensor networks, spatiotemporal data models, decentralized algorithms, qualitative spatial changes.

1 Introduction

Many spatial phenomena are continuously changing in time and space [5]. Environmental monitoring, for example, focuses especially on areas of environmental disturbance, such as grazing, fertilizing, pollution, and logging. Environmental disturbance is one of the key driving forces for environmental changes, and different disturbance events can result in diverse changes. For example, a logging event can result in the decrease of forest coverage and the change from forest to heath land. A flooding event can result in the increase of soil moisture and the transform from heath land to wetland [9].

Wireless sensor networks (WSNs) are increasingly being used in spatial applications to detect change in the environments. A WSN is a network of computing devices that can collaborate via radio communications. A WSN is also a network of observation devices, since each node in the WSN is equipped with sensors that enable thematically fine-grained observation of the environments. A WSN is able to observe change in real-time, and hence, with almost any temporal granularity (but typically coarser spatial granularity).

In the context of geographic information science, snapshot-based and event-based approaches have been used to model wireless sensor networks. Snapshot-based approach are commonly used in applications, where sensor nodes are tasked to periodically sense and transmit snapshots of an environment by setting the WSN to a certain temporal sensing resolution (e.g., [16]). Although the

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 219–233, 2010.

snapshot approach is more practical, theoretical studies (e.g., [18]) are more interested in event-based model of WSNs due to its advantages of detecting salient changes or events.

However, practically also event-based models are subject to granularity effects. By certain spatial and temporal resolution of observation, events may not be recorded somewhere, sometime. Even if someone may argue that continuous observation is possible with future technologies, there is still the problem of redundancy of observations. Large amount of energy can be consumed to observe an environment where no change has occurred. In this circumstance, this paper develops a new spatiotemporal data model that incorporates both snapshot-based and event-based approaches, such that the model can adapt to different granularities of observations, and can detect and analyze spatial changes.

In our spatiotemporal data model, a WSN is modeled as a set of point objects with point observations, and the primary structure in our model are edges. Edges are important components for spatial objects. For example, an area is embedded in a polygon that is constituted as a sequence of edges. By sensor network point observations, geographic phenomena will be abstracted as areal objects that consist of points, and these points will then be organized as sequence of edges. Since temporal granularity of observation is allowed in our model, geographic phenomena are represented as sequences of snapshots. Our model supports the analysis of change in sequences of snapshots that is captured by different granularity of observations. Figure 1 illustrates an example about the representation of dynamic areal objects in a sensor network.

This paper has four major contributions. Firstly, we propose a new data structure that ensures the boundaries of areal objects are always closed traversable trails. Secondly, we propose deleted and inserted edges to represent changes of areal objects in snapshot sequences. The sequences of deleted and inserted edges always form closed traversable trails, and there are eight different types of these closed trails that can be used to distinguish six topological changes and two non-topological changes. Thirdly, our approach can support both incremental and non-incremental changes. And finally, we develop a decentralized algorithm to efficiently detect the eight different types of changes in a sensor network.

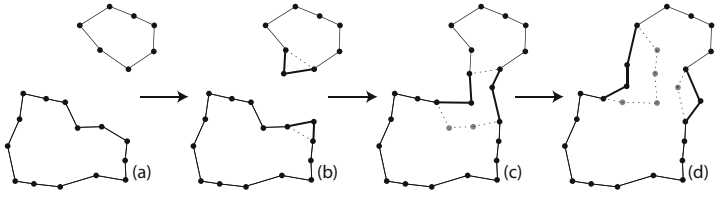


Fig. 1. The evolving of spatial objects over time. (a)-(d) are at different time steps, e.g., t_1, t_2, t_3, and t_4.

2 Related Work

The change investigated in this paper is the change of spatial entities. A change occurs whenever spatial entities possess different spatial attributes at different times [2]. Grenon and Smith [4] classify entities in the spatiotemporal world into two categories: one is continuants, and the other is occurrents. The study of this paper is on the perspective of continuants that exist at a given time at a given level of granularity and undergo different types of changes over time. Hornsby and Egenhofer [5] suggest a qualitative representation of change. Their notion of change is based on object identity, and a set of operations that either preserve or change identity. This paper applies their concept of identity assuming that sensor network can be set up with a certain appropriate granularity of observations for a certain environmental phenomenon, according to the sampling theorem [14].

Grenon and Smith [4] propose SNAP and SPAN ontologies. The SNAP ontology has a snapshot-based view, where entities are organized as temporal sequences of snapshots. On the other hand, event-based spatiotemporal data models in SPAN ontology (e.g., [10,11,17]) can be understood as a chronicle-based model that is dual to the snapshot-based model [3]. Snapshot-based and event-based approaches are commonly used in wireless sensor network applications.

Lian *et al.* [6] propose a gradient boundary detection approach, where snapshots of a monitored region are transmitted to a designated central computer via the *sink* of the network at certain time intervals. A snapshot at a given time t is abstracted as a contour map, which consists of gradient boundaries. Only the nodes on the gradient boundaries need to report to the sink, which constructs a contour map. A unique work for collecting snapshot data is proposed by Skraba *et al.* [15]. Their method is called a sweep, i.e., a wavefront that traverses the whole network and passes all nodes in the network exactly once. These approaches are limited to static snapshots. They cannot dynamically analyze snapshot sequences to derive salient changes.

Duckham *et al.* [1] develops a model to track salient changes or events in the environment. They model a sensor network as a triangulation network. The triangulation can change over time in response to the movement of spatial phenomena. Worboys and Duckham [18] also use triangulations, and spatial objects are represented as a set of triangles. Changes of spatial objects are represented as the insertion or deletion of triangles from the set. However, their approach is limited to incremental changes, i.e., the change of a single triangle at each time step. Sadeq and Duckham [13] investigate different sensor network structures for detecting topological changes. Several commonly-used neighborhood structures, such as Delaunay triangulation, Gabriel graph, relative neighborhood graph, and greedy triangulation, have been tested. Their approach also assumes incremental changes. Their classification of spatial changes is based on the theoretical study of Jiang and Worboys [8].

A large amount of existing work on qualitative approaches to characterize topology and topological changes in sensor network exists, e.g., [7]. This paper is distinguished from other related work by using edge sequences for qualitative reasoning. We use deleted and inserted edges to represent changes of areal

objects, and the sequences of deleted and inserted edges are always closed trails regardless the number of nodes on the trails. Thus, our approach allows for the first time both incremental and non-incremental changes.

3 Change Representation in Sensor Network

A sensor network can be modeled as a directed planar graph $G = (V, E)$, which is built by Delaunay triangulation [12]. In this paper we assume that the sensor network is dense enough to ensure the construction of a triangulation framework. In the graph G, V is a set of nodes and E is a set of edges, e.g., (v, v'), which represent the direct communication links between the nodes $v, v' \in V$. We assume that E is *symmetric*, i.e., if $(v, v') \in E$, then $(v', v) \in E$. Note that the direction of a directed edge will be illustrated in a figure when it is relevant, otherwise the representation of G can be simplified as in Figure 2(a).

The set of neighbors of $v \in V$ is denoted as $nbr(v) = \{v' : (v, v') \in E\}$. For example in Figure 2(a), $nbr(v) = \{a, b, c, d, e\}$, and the set $nbr(v)$ is sorted into clockwise order. Each node only stores data about itself and its immediate neighbors.

The set of directed edges adjacent to a node v is denoted as $edge(v)$. Since E is symmetric, the set $edge(v)$ can be derived from $nbr(v)$, i.e., $edge(v) = \{(v, v') : v' \in nbr(v)\} \cup \{(v', v) : v' \in nbr(v)\}$. In Figure 2(b), for example, the node v has five neighbors, so that v has ten directed edges. The set $edge(v)$ is also sorted in clockwise order, and for a same neighboring node v', the edge (v', v) is always the next edge of the edge (v, v'), as in Figure 2(b).

A node v and its neighbors are organized into *directed cycles*, or simply *cycles*. A cycle is denoted as (v, b, a, v), where the three nodes v, a and b are distinct, and there is an edge for any two consecutive nodes in the cycle, i.e., (v, b), (b, a), $(a, v) \in E$. Since $nbr(v)$ is in clockwise order, all the cycles are counterclockwise, and an example, i.e., (v, e, d, v), is shown in Figure 2(b). Based on the above definition, we can define the representation of areal objects in a sensor network:

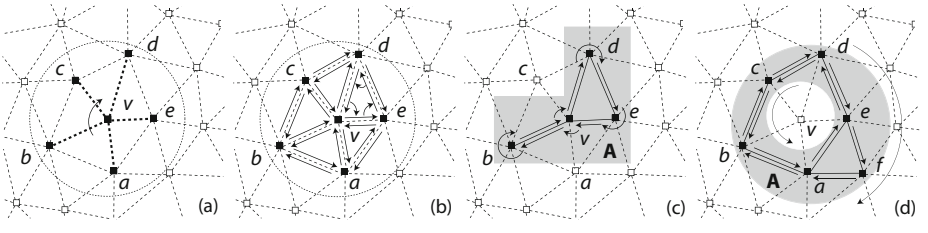


Fig. 2. Sensor network structure. (a) The set $nbr(v)$ of each node v is sorted into clockwise order. (b) The directed edges and cycles of a node v. (c) The traversal of boundary edges based on the clockwise rule. (d) A region with a hole can be detected by the traversal orientations.

Definitions: A directed edge $(v, v') \in E$ is an *object edge*, if both v and v' are located in an areal object. A cycle (v, b, a, v) is an *object cycle*, if all the vertexes of the cycle, i.e., v, a, b, are in an areal object. An object edge $(v, v') \in E$ is a *non-boundary edge*, if the edge belongs to an object cycle. An object edge $(v, v') \in E$ is a *boundary edge*, if it does not belong to any object cycle.

Areal objects can be represented as sequence of boundary edges. An example is shown in Figure 2(c), in which the nodes b, d, e and v are located in an areal object A. The edges (b, v), (v, d), (d, e), (e, v), (v, b) are boundary edges, since they do not belong to any object cycles. Usually only boundary edges will be illustrated in the figures, as in Figure 2 (c) and (d).

Traversals on boundary edges of areal objects always form closed trails. For example, in Figure 2(c), suppose the traversal is started at node b, then the traversal will be a closed trail followed the clockwise rule: $b \rightarrow v \rightarrow d \rightarrow e \rightarrow v \rightarrow b$. Formally, the path of a set of connected boundary edges is a *Eulerian trail*, since the number of boundary edges in each node is always even.

An important property of these trails is the traversal orientation. Given a region with a hole, for example in Figure 2(d), external traversals of the region are clockwise and internal traversals of the hole are counterclockwise. Since the area of a polygon is positive if the vertexes of the polygon are arranged in a counterclockwise order, and negative if they are in clockwise order, the result of area calculation can be used to determine the ordering of the vertices of a polygon, and thus the orientation of a traversal.

In our model, the change of areal objects will be captured as sequences of snapshots in the sensor network. In each snapshot, sequences of boundary edges are used to represent areal objects. Since areal objects are evolving over time, the sequences of boundary edges would dynamically change at different snapshots, as in Figure 3(a)-(c). The change of boundary edges between two consecutive snapshots is represented by the *insertion* and *deletion* of boundary edges:

Definitions: A boundary edge $(v, v') \in E$ is an *inserted boundary edge*, or simply an *inserted edge*, at time t_i if the edge is not a boundary edge at previous time t_{i-1}, but become a boundary edge at time t_i. A boundary edge $(v, v') \in E$ is a *deleted edge* at time t_i if the edge is a boundary edge at previous time t_{i-1}, but is not a boundary edge at time t_i.

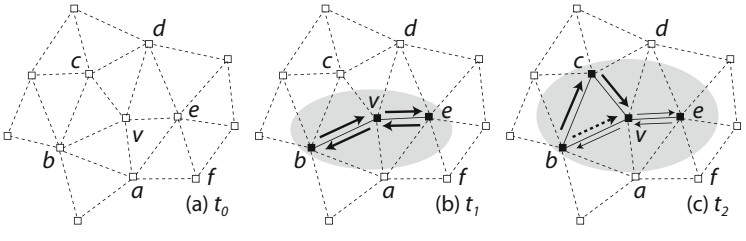


Fig. 3. Inserted and deleted boundary edges are marked by thick solid-line arrows and thick dashed-line arrows respectively. Only boundary edges are illustrated in the figure.

We assume the sensor reading is binary, i.e., $\{0, 1\}$, where the reading of 1 represents that the sensor node is located in an areal object. We define that if a node v changes its sensor readings, e.g., from 1 to 0, at time step t_i, then v is an active node at t_i. For example, in Figure 3, nodes b, e, and v are active nodes at t_1, and node c is the only active node at t_2. We define that between two consecutive time steps t_i and t_{i-1}, a change of areal object is incremental if there is one and only one active node at time step t_i. In Figure 3, the change between t_0 and t_1 is non-incremental and the change between t_1 and t_2 is incremental. Our model supports both incremental and non-incremental changes.

4 Detecting Change

The traversals on deleted and/or inserted edges can be used to distinguish eight different types of changes of areal objects. These eight types of changes include six topological changes and two non-topological changes. The six topological changes are appearance, disappearance, merge, split, self-merge and partial-split [8,13]. The two non-topological changes are expansion and contraction. Examples of the eight types of changes are shown in Figure 4.

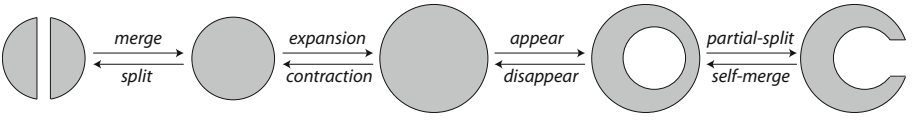


Fig. 4. Six topological changes and two non-topological changes

As will be discussed, different types of changes will have different types of traversals on deleted/inserted edges, and all of the traversals will form closed traversable trails. We discuss the eight types of changes in four groups.

4.1 Appearance and Disappearance

There is a closed traversable trail for the appearance of areal objects. In Figure 5(a), an areal object appears, and a set of boundary edges has been inserted. The set of inserted boundary edges forms a closed trail. Suppose the traversal starts at node i, then the trail is: $i \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h \rightarrow i$. An abstract representation of the traversal in Figure 5(a) is shown in (b), where the solid-line arrow represents a traversal on inserted edges, and the circle represents a node that has a node identity of i. Note that Figure 5 illustrates the stage when changes have occured but areal object identities have not yet been updated.

In Figure 5(c), an areal object disappears, and all the boundary edges with areal object identity of A have been deleted. And in Figure 5(d), the dashed-line arrow denotes a traversal on deleted edges. Note that areal object identity is decentrally stored in each boundary edge of the areal object.

4.2 Expansion and Contraction

For the expansion and contraction of areal objects, the closed trail will include two *segments*. One segment of the traversal consists of inserted edges, and another segment consists of deleted edges. In Figure 5(e), for example, an areal object expands. The inserted edges form one segment of the traversal: $b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h$. The deleted edges form another segment: $h \rightarrow j \rightarrow b$. Notice that the second segment of the traversal is in reverse direction. Figure 5(f) illustrates that the two segments of the traversal are connected by two nodes b and h. The two nodes b and h are called the *transition nodes* of the traversal. Segment 2 is aware the areal object identity A, while Segment 1 do not have areal object identity yet.

Similarly, in Figure 5(g) and (h), an areal object contracts, and there are two segments that is connected by two transition nodes b and h. Section 5 will demonstrate how expansion and contraction can be distinguished: the traversal of expansion consists of (1) *inserted edges* and (2) *deleted edges*, and the traversal of contraction consists of (1) *deleted edges* and (2) *inserted edges*.

4.3 Merge and Split

The closed trail for the merge of two areal objects includes four segments. Figure 5(i) shows a merge of two areal objects A and B. The first segment of the traversal is on inserted edges from node a to node d: $a \rightarrow b \rightarrow c \rightarrow d$. Two transition nodes a and d are located in different areal objects, i.e., A and B respectively. The second segment of the traversal is on the deleted edges of areal object B from node d to node m: $d \rightarrow e \rightarrow m$, and the deleted edges have the areal object identity B, as shown in Figure 5(i) and (j). The third segment will start from areal object B and return to areal object A via inserted edges: $m \rightarrow f \rightarrow g$. Finally, the fourth segment of the traversal will return to the original node a via the deleted edges of areal object A: $g \rightarrow j \rightarrow a$. Note that Segments 2 and 4 are in reverse direction. These four segments of traversal are connected by four transition nodes a, d, m, and g, in which two nodes are located in areal object A, and the other two are in areal object B.

The traversal of a split also includes four segments, and the four segments are in the sequences of (1) *deleted edges* and (2) *inserted edges* and (3) *deleted edges* and (4) *inserted edges*, as in Figure 5(k) and (l). The four segments of traversal are also connected by four transition nodes a, d, m, and g. Segments 1 and 3 on deleted edges are aware the areal object identity A. In comparison, the four traversal segments of a merge are in a different order: (1) *inserted edges* and (2) *deleted edges* and (3) *inserted edges* and (4) *deleted edges*, as in Figure 5(j). Both of the traversals in Figure 5(j) and (l) start at node a. The selection of the traversal starting node will be discussed in Section 5.

4.4 Self-merge and Partial-Split

In Figure 5(m), an areal object A self-merges into a region with a hole. The traversal of self-merge has a similar structure as a merge: (1) *inserted edges*

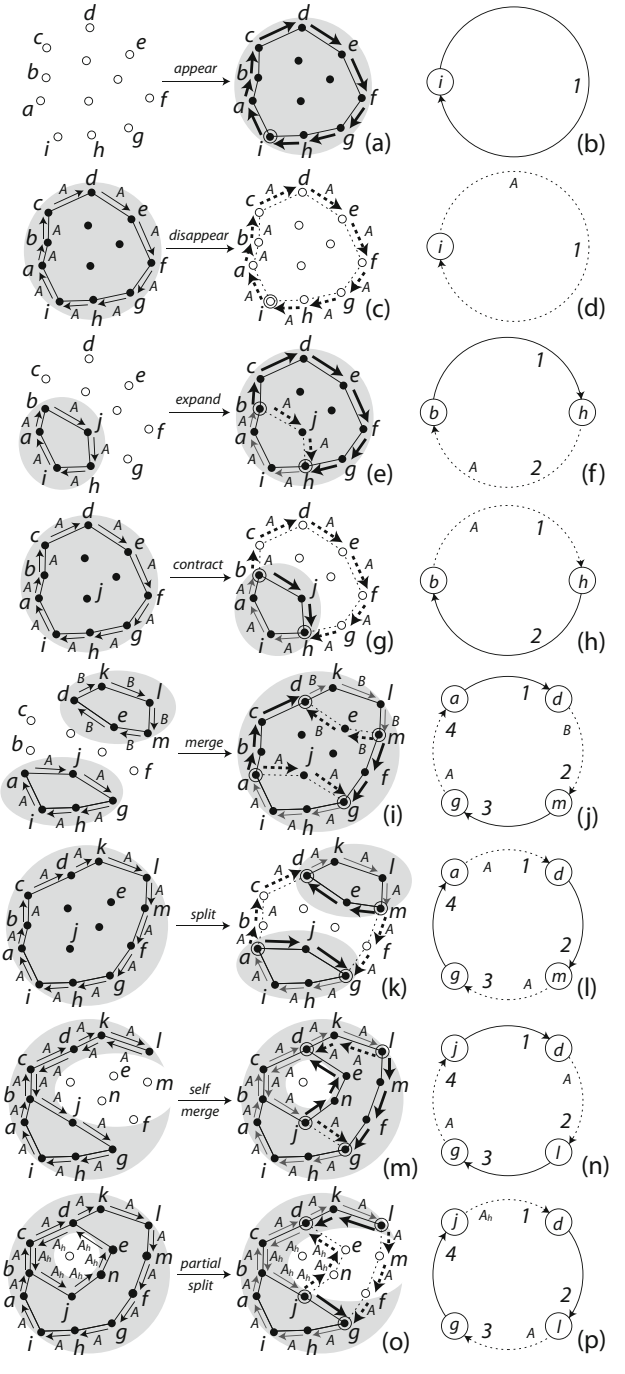


Fig. 5. Eight traversals form eight different closed traversable trails

and (2) *deleted edges* and (3) *inserted edges* and (4) *deleted edges*, as shown in Figure 5(n). Segments 2 and 4 of a self-merge have the same areal object identity A, as illustrated in Figure 5(m) and (n). In comparison, Segment 2 and 4 of a merge in Figure 5(i) and (j) have different areal object identities, i.e., B and A respectively. Thus, merges and self-merges can be distinguished by the areal object identities of the traversal segments, as in Figure 5(j) and (n).

In Figure 5(o), an areal object with a hole partial-splits. The areal object with a hole consists of two close trails: one is internal for the hole and the other is external for the areal object, as already introduced in Section 3. The areal object identity for a hole of an areal object A is specified as A_h. Figure 5(o) shows that all the internal edges with A_h form a closed trail: $b \rightarrow j \rightarrow n \rightarrow e \rightarrow d \rightarrow c \rightarrow b$. Similar to splits, the traversal for partial-split has four segments, and the order is (1) *deleted edges* and (2) *inserted edges* and (3) *deleted edges* and (4) *inserted edges*. For a split, Segments 1 and 3, in Figure 5(l), have the same areal object identity A, while in the case of partial-split in Figure 5(p), Segment 1 has an areal object identity of A_h and Segment 3 has a different identity A.

4.5 Summary

As a summary, there are eight different types of basic traversals on deleted and inserted edges. These eight traversals can be distinguished based on their segments. Each type of traversal can uniquely identify one type of change. There are possible other types of traversals, and they can be considered as the combinations of the eight basic traversal types. For example, in Figure 6, an areal object expands, and the deleted and inserted edges have been separated into two closed trails: i.e., $d \rightarrow o \rightarrow n \rightarrow d$ and $i \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h \rightarrow i$. The traversals in Figure 6 can be considered as a combination of the two traversals in Figure 5(b) and (d). Our algorithms are able to handle the combination of the eight basic traversals such that different change types can be identified.

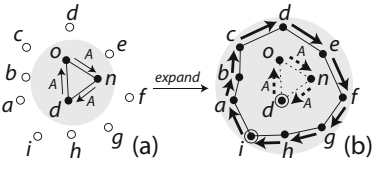


Fig. 6. Other types of traversals can be considered as the combinations of the eight basic traversal types. The traversal in (b) is a combination of two traversals in Figure 5(b) and (d).

5 Algorithm

As discussed in Section 4, there are eight different types of traversals that can be used to distinguish eight types of changes. This section will provide decentralized algorithms for traversal organization and change detection. The basic idea of our

algorithm is to initialize a traversal at a given node v with a *message*, and the message will be passed from one node to another during the traversal. Also, decentralized data will be aggregated into the message during the traversal. Since all the traversals form closed trails, the aggregated message will return to its origin node v, so that node v will be able to detect different types of changes based on the aggregated message.

5.1 Messages

Since areal object identities of traversal segments can be used to identity different types of changes as in Figure 5, relevant areal object identities will be aggregated during the traversal. A message will be initialized at the beginning of a traversal for data aggregation in the network. A message initialized at a node v is denoted as: $msg(v) = (p_1, p_2, p_3, p_4)$.

The four elements, i.e, p_1, p_2, p_3, and p_4, denote the areal object identities of the four segments in a traversal. We use $msg(v).p_1$, for example, to represent the element p_1 in $msg(v)$. In the beginning of a traversal, all the elements in a message are empty, and the message can be represented as: $msg(v) = (\emptyset, \emptyset, \emptyset, \emptyset)$.

The transition nodes are responsible to update the elements of a message during a traversal. For example, the closed trail for a split in Figure 5(k) consists of four transition nodes a, d, m, and g, and four segments. If node a initializes a traversal with a message $msg(a)$, then node a will update the element $msg(a).p_1$, and d, m, and g will update $msg(a).p_2$, $msg(a).p_3$, and $msg(a).p_4$ respectively. Finally, the traversal will return to the initial node a with a message that contains all the information about the closed trail.

All the edges in the same segment have the same areal object identity. For example, in Figure 5(k), the edges (a, b), (b, c) and (c, d) in the first segment have the same identity of A, and thus $msg(a).p_1 = A$. Similarly, $msg(a).p_2 = I$, $msg(a).p_3 = A$, and $msg(a).p_4 = I$. Note that $msg(a).p_2 = I$ and $msg(a).p_4 = I$ are used to specify that the second and fourth segments of the traversal are on *inserted* edges and areal object identities have not been provided. In Figure 5(k), a completed message for the traversal started at node a would be: $msg(a) = (A, I, A, I)$.

5.2 Absent Traversal Segments

In some traversals, it is possible that there is no edge in a segment, and the segment is regarded as an *absent* segment. For example, in Figure 7(b), if node b starts a traversal, then the traversal would only contains three segments: $b \rightarrow c \rightarrow d$ on inserted edges, $d \rightarrow c \rightarrow a$ on inserted edges, and $a \rightarrow b$ on a deleted edge. Since a traversal of merge has (1) *inserted edges* and (2) *deleted edges* and (3) *inserted edges* and (4) *deleted edges*, as in Section 4.3, we regard the second segment on deleted edges as an absent segment at node d. The areal object identity for the absent segment should be provided by node d. As shown in Figure 7(b), node d can acquire the areal object identity, i.e., B, from its boundary edges $\{(d, e), (e, d)\}$. The completed message initialized by node b would be: $msg(b) = (I, B, I, A)$.

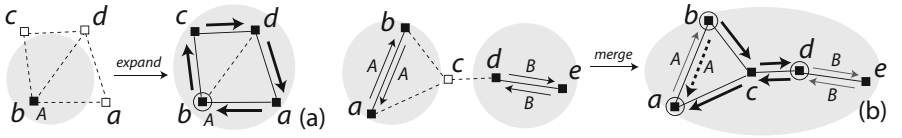


Fig. 7. It is possible that there are absent segments in a traversal. Transition nodes are marked by circles. (a) A segment on deleted edges is absent at node b. (b) A segment on deleted edges is absent at node d.

Generally, in traversals of merge, split, self-merge and partial-split, the first and third segments always exist, while the second and fourth segments may be absent. In the traversals for expansion and contraction, the first segment always exists, while the second segment may be absent, as an example in Figure 7(a).

5.3 Transition Edges

Since our algorithm should be able to efficiently organize traversals for decentralized data aggregation, only a small subset of sensor nodes should be nominated for initializing traversals. We define that an inserted or deleted edge is a *transition edge* if the edge connects an active node and a non-active node.

For example, in Figure 5(k), the deleted edge (a, b) is a transition edge that has a initial non-active node a and a terminal active node b. Also (c, d) is a transition edge, but the edge begins at an active node c and ends at a non-active node d.

Transition edges are always in pairs. If there is a transition edge with an initial non-active node and a terminal active node, then there always exists another transition edge with an initial active node and a terminal non-active node. A pair of transition edges is always connected by a trial. For example, in Figure 5(k), (a, b) and (c, d) are connected by the trial $a \rightarrow b \rightarrow c \rightarrow d$.

If a node v is a non-active node and is the initial node of a transition edge, then node v has the responsibility to initialize a traversal (*type 1*) with a message $msg(v)$. For example, in Figure 5(k), node a is a non-active node, and it is the initial node of a transition edge (a, b), so a should start a traversal with a new message $msg(a)$. The traversal will visit nodes b, c, d, e, m, f, g, and j, and return to the starting node a with a completed message $msg(a)$. Similarly, in Figure 5(k), the non-active node m is the initial node of (m, f), and thus node m should also initialize a traversal.

In the case of appearance or disappearance of an areal object, all the nodes in the areal object are active nodes, and thus there is no transition edge. Another type of traversal (*type 2*) is required for these cases. Although our algorithm includes the organization and combination of traversals *type 1* and *type 2*, the details are not further discussed.

Algorithm 1: Detecting Change

```
1  Variables: node v;
2  if v receives a message msg(v) initialized by itself then
3      if msg(v).p1 ≠ ∅ and msg(v).p2 = ∅ then
4          if msg(v).p1 = I then v detects an appearance;
5          if msg(v).p1 ≠ I then v detects a disappearance;
6      else if msg(v).p2 ≠ ∅ and msg(v).p3 = ∅ then
7          if msg(v).p1 = I then v detects an expansion;
8          if msg(v).p1 ≠ I then v detects a contraction;
9      else if msg(v).p4 ≠ ∅ then
10         if msg(v).p1 = I then
11             if msg(v).p2 ≠ msg(v).p4 then v detects a merge;
12             if msg(v).p2 = msg(v).p4 then v detects a self-merge;
13         else if msg(v).p1 ≠ I then
14             if msg(v).p1 ≠ msg(v).p3 then v detects a partial-split;
15             if msg(v).p1 = msg(v).p3 then v detects a split;
```

5.4 Detecting Change

If a node v receives a message $msg(v)$ initialized by itself, then node v is able to detect different types of changes based on the message $msg(v)$ (see Algorithm 1, line 2). If the message contains only one segment, i.e., $msg(v).p_1 \neq \emptyset$ and $msg(v).p_2 = \emptyset$, then node v detects an appearance or a disappearance (lines 3-5). If the segment is on inserted edges, i.e., $msg(v).p_1 = I$, then node v detects an appearance (line 4). If the segment is on deleted edges, i.e., $msg(v).p_1 \neq I$, then node v detects a disappearance (line 5).

If the message contains two segments, i.e., $msg(v).p_2 \neq \emptyset$ and $msg(v).p_3 = \emptyset$, then node v would detect an expansion or contraction (lines 6-8). As illustrated in Figure 5(f) and (h), the first segment of expansion is on inserted edges (line 7), while the first segment of contraction is on deleted edges (line 8).

There are four types of changes, i.e., merge, split, self-merge, and partial-split that have four traversal segments (lines 9-15). In the cases of merge and self-merge, the first segment is on inserted edges (lines 10-12). Merge and self-merge can then be distinguished by the second and fourth segments of traversals. The second and fourth segments have different areal object identities for a merge, and have the same areal object identity for a self-merge. While for split and partial-split, the first segment is on deleted edges (lines 13-15). The first and third segments have the same areal object identity for a split, and have different areal object identities for a partial-split.

6 Evaluation

The decentralized algorithm in Section 5 was evaluated in a simulation environment. Repast (http://repast.sourceforge.net/) was used for simulation. The

directed planar graph structure was built by Delaunay triangulation in Repast. The planar graph consists of 500 sensor nodes that are deployed in a region of 800×800 square units, with a communication range of 80 units.

In the experiment, we had about 2000 simulation runs, and each type of change had about 250 runs. An areal object is simulated as one or several $r \times r$ square boxes, as examples in Figure 8. From the 1st to the 2000th simulation runs, the parameter r gradually increase, and thus the number of active nodes should also gradually increase. At each simulation run, a specific type of spatial change can occur at randomized locations within the sensor network. Figure 8 (a) and (b) show examples of merge and partial-split that had occured at different locations of the sensor network.

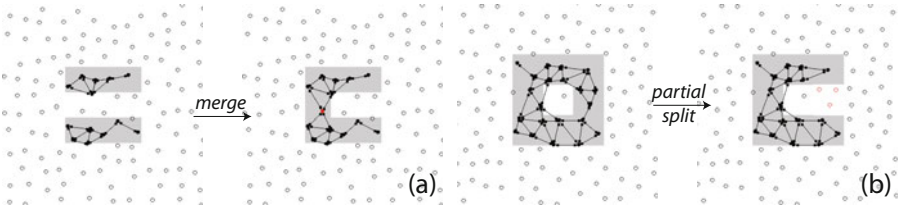
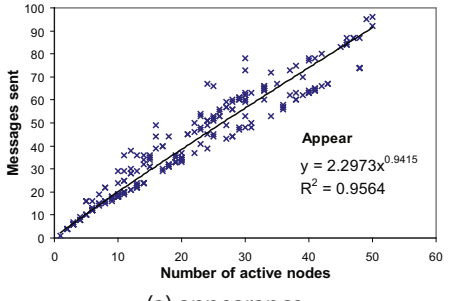
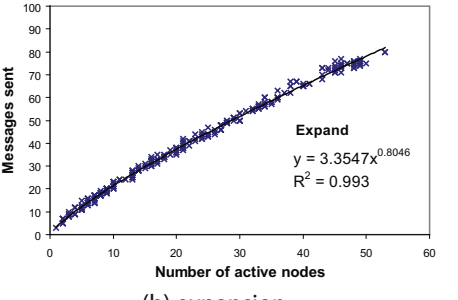


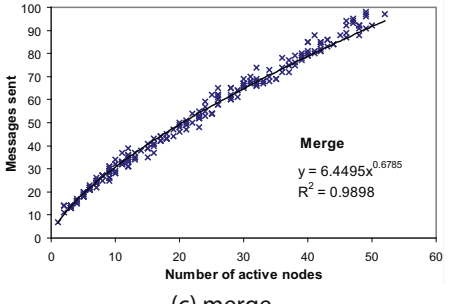
Fig. 8. Spatial changes occur at randomized locations within a sensor network



(a) appearance



(b) expansion



(c) merge

Types of change	*a*	*b*	R^2
Appearance	2.2973	0.9415	0.9564
Disappearance	2.3347	0.9353	0.9551
Expansion	3.3547	0.8046	0.9930
Contraction	3.4046	0.8014	0.9926
Merge	6.4495	0.6785	0.9898
Split	7.0449	0.6552	0.9877
Self-merge	6.4553	0.6934	0.9893
Partial-split	6.2780	0.6975	0.9896
Centralized	11.6970	1.1079	0.9984

(d) regression analysis $y = a \cdot x^b$

Fig. 9. Scalability of the algorithm

Our algorithms were installed in each sensor node in the network. In all the simulation runs, the eight different types of changes were correctly detected by the directed planar graph structure, as long as the granularity of observation is set up appropriately based on the granularity of areal objects.

The experiment investigated the scalability of the algorithm. The scalability was measured by number of messages sent with increasing number of active nodes. The results of power regression analysis for appearance, expansion, and merge, for example, is shown in Figure 9(a), (b), and (c) respectively. Figure 9(d) lists the results of regression analysis for the eight types of changes. A power regression ($y = a \times x^b$) is used to fit the 250 plotted results for each type of change. All the regression results achieve high goodness of fit, with the range of R^2 from 0.9551 to 0.993. As in Figure 9(d), there is $b < 1$ for all the eight types of changes, and thus our algorithm should have an order of $O(n)$ or less. The experiment shows that our decentralized algorithm is highly scalable.

To compare with our decentralized algorithm, we also implement a centralized sense-and-transmit algorithm, in which each active node will simply forward its sensor reading to a sink by multi-hop routings. The results of the centralized algorithm is shown in the last row of Figure 9(d). By comparing a and b in $y = a \times x^b$, it is clear that our decentralized algorithm is more message efficient and scalable than a centralized algorithm.

7 Conclusion

In this paper, we have developed a new spatiotemporal data model that enables the detection of qualitative spatial changes in snapshot sequences. In our model, boundaries of areal objects are closed traversable trails. Eight types of changes of areal objects can also be represented and distinguished by eight different closed trails that consist of inserted and/or deleted edges. We have also developed a decentralized algorithm for WSNs to detect changes by in-network aggregations. The experiment proves that our algorithm is scalable and efficient, and is able to detect changes with different numbers of active nodes. In the future work, our data model can be further developed to support decentralized spatial queries, for example, the query of the topological relations between two spatial regions.

References

1. Duckham, M., Nittel, S., Worboys, M.F.: Monitoring dynamic spatial fields using responsive geosensor networks. In: Proceedings of 13th Annual ACM International Workshop on Geographic Information Systems (GIS 2005), pp. 51–60 (2005)
2. Galton, A.: Qualitative Spatial Change. Oxford University Press, Oxford (2000)
3. Galton, A.: Fields and objects in space, time, and space-time. Spatial Cognition and Computation 4(1), 39–67 (2004)
4. Grenon, P., Smith, B.: SNAP and SPAN: Towards dynamic spatial ontology. Spatial Cognition and Computation 4(1), 69–104 (2004)
5. Hornsby, K., Egenhofer, M.: Qualitative Representation of Change. In: Frank, A.U. (ed.) COSIT 1997. LNCS, vol. 1329, pp. 15–33. Springer, Heidelberg (1997)

6. Lian, J., Chen, L., Naik, K., Liu, Y., Agnew, G.B.: Gradient boundary detection for time series snapshot construction in sensor networks. IEEE Transactions on Parallel and Distributed Systems 18(10), 1462–1475 (2007)
7. Jiang, J., Worboys, M.: Detecting basic topological changes in sensor networks by local aggregation. In: ACM GIS, Irvine, CA, USA (2008)
8. Jiang, J., Worboys, M.: Event-based topology for dynamic planar areal objects. International Journal of Geographical Information Science 23(1), 33–60 (2009)
9. Mau, I., Hornsby, K., Bishop, I.D.: Modeling geospatial events and impacts through qualitative change. In: Barkowsky, T., Knauff, M., Ligozat, G., Montello, D.R. (eds.) Spatial Cognition 2007. LNCS (LNAI), vol. 4387, pp. 156–174. Springer, Heidelberg (2007)
10. Peuquet, D.J., Duan, N.: An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. International Journal of Geographical Information Systems 9, 7–24 (1995)
11. Peuquet, D.J.: Making Space for Time: Issues in Space-Time Data Representation. Geoinformatica 5(1), 11–32 (2001)
12. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer, New York (1985)
13. Sadeq, M.J., Duckham, M.: Effect of neighborhood on in-network processing in sensor networks. In: Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 133–150. Springer, Heidelberg (2008)
14. Shannon, C.E.: Communication in the presence of noise. Proceedings of the Institute of Radio Engineers 37(1), 10–21 (1949)
15. Skraba, P., Fang, Q., Nguyen, A., Guibas, L.: Sweeps over wireless sensor networks. In: IPSN 2006: Proceedings of the 5th International Conference on Information Processing in Sensor Networks, pp. 143–151 (2006)
16. Wark, T., Corke, P., Sikka, P., Klingbeil, L., Guo, Y., Crossman, C., Valencia, P., Swain, D., Bishop-Hurley, G.: Transforming Agriculture through Pervasive Wireless Sensor Networks. IEEE Pervasive Computing 6(2), 50–57 (2007)
17. Worboys, M.: Event-oriented approaches to geographic phenomena. International Journal of Geographical Information Science 19(1), 1–28 (2005)
18. Worboys, M., Duckham, M.: Monitoring qualitative spatiotemporal change for geosensor networks. International Journal of Geographical Information Science 20(10), 1087–1108 (2006)

Multi-source Toponym Data Integration and Mediation for a Meta-Gazetteer Service

Philip D. Smart, Christopher B. Jones, and Florian A. Twaroch

School of Computer Science, Cardiff University
{c.b.jones,p.d.smart,f.a.twaroch}@cs.cf.ac.uk

Abstract. A variety of gazetteers exist based on administrative or user contributed data. Each of these data sources has benefits for particular geographical analysis and information retrieval tasks but none is a one fit all solution. We present a mediation framework to access and integrate distributed gazetteer resources to build a meta-gazetteer that generates augmented versions of place name information. The approach combines different aspects of place name data from multiple gazetteer sources that refer to the same geographic place and employs several similarity metrics to identify equivalent toponyms.

Keywords: gazetteers, place names, spatial data integration, mediation architecture, geo-web services.

1 Introduction

Automated gazetteer services that maintain geo-data associated with geographic place names are becoming increasingly important for a variety of applications [1]. They are needed to recognise place names that users employ in queries to retrieve geographic information and for applications that need to detect the presence of place names in text resources, for example to index documents for a spatially-aware search engine. Gazetteer services are also required for the reverse geocoding process of finding place names associated with geographical coordinates, e.g. to attach a place name to a GPS-referenced photo. To provide effective support for these sorts of applications raises the challenge of creating a gazetteer that can maintain access to a wide range of place name terminology relating to many different sorts of features at arbitrary locations. In practice however, because the quality of the content of a gazetteer will depend upon the application for which it is required [2], it is not possible to specify the characteristics of a single ideal gazetteer, except perhaps at a generic level.

At present the number and types of gazetteer resources are increasing as commercial gazetteers become supplemented by volunteered sources of geographic place name data. The gazetteer sources differ considerably with regard to their geographical coverage, the range of features types, the presence of alternative names, and the detail and accuracy of their geometric footprints. National mapping agencies (NMA) generate gazetteers that relate typically to their respective geographical areas, and while they may be reasonably reliable with regard to representation of formal administrative geographic entities, their range of types of named feature and of the terminology employed is inevitably limited to particular, usually topographic, themes. Commercial

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 234–248, 2010.

sources of place name knowledge may have wider geographical extents but are usually focused on particular types of application, especially that of navigation, and tend to reflect the administrative view of geography. Volunteered place name resources, such as the Geonames gazetteer and OpenStreetMap may support a range of features types and levels of detail not found within the commercially-marketed and NMA sources. It is also the case that gazetteer sources differ in their data structures and access methods. Some need to be loaded to a database, while others have web service or other interfaces.

Given these varied sources of gazetteer knowledge, there is a motivation to create a "meta-gazetteer" that accesses multiple resources in order to retrieve the best toponym information available for a particular purpose. Because the different sources of place name information differ in the quality of their associated data, such as the feature types and the spatial footprints, there is need for a form of conflation [3] in which multiple sources are compared and merged so that the best aspects of each source can be combined. In the present paper we address these requirements and describe methods for multi-source place name data integration and mediation that have been developed to support a distributed web gazetteer service, that is used for geo-parsing free text and for reverse geo-coding. After a review of related work in section 2 we give an overview of our Toponym Ontology data model and the associated mediation-based architecture (section 3). Section 4 analyses the characteristics of the formal and volunteered sources employed and section 5 explains our mediation system methods for selecting, integrating and augmenting toponym geofeatures from multiple resources. We present results and an evaluation of the geofeature matching procedure in section 6 and give then an outlook to future work.

2 Related Work

Typical data that are stored in gazetteers are standard and alternative names of a geofeature, the type of the named feature, a geometric coordinate-based footprint, such as a point, a bounding box or a polygon, and one or more parent features within an administrative or topographic hierarchy [1]. Gazetteer specifications such as that of the Alexandria Digital Library support further attributes such as spatial and other relations between features, the data accuracy, and the source of the data for an individual place.

In a recent review of requirements for a next generation gazetteer, Keßler et al [4] drew attention to a number of desirable gazetteer properties which include those of accessing multiple data sources, exploiting volunteered data sources, maintaining mechanisms to assess trust in resources, development of an agreed high level domain ontology, inclusion of deductive inference of knowledge and the development of a semantically enabled user interface.

Gazetteer enrichment from web resources still faces a number of challenges: for example, place name recognition methods applied to text corpora degrade significantly in their performance (measured in recall and precision of developed GIR systems) when no gazetteer is initially considered [5], i.e. to construct a gazetteer we need a gazetteer. Uryupina [6] presented a method to detect place names together with their feature type using a search engine and machine learning techniques but does not

focus on a particular geographic region. Natural language processing methods to detect place names in text corpora do not ground them with geometric footprints.

Goldberg et al [2] created enriched name and feature type data for merged address (parcel) level places using multiple representations of the same place derived from online residential and commercial phone books. Equivalence of features was established in terms of equivalence of the name and address attributes. This simple testing was facilitated by a prior data cleansing or normalisation process that transformed their sources to a common USPS address format, using the probabilistic "record linkage" methods of Christen and Churches [7]. A third "official" county web site data source was used to validate derived addresses.

Flickr tags have been used to build representations of place [8, 9]. While this is a step in the right direction the approaches depend heavily on the availability of volunteered geographic information in a single source. Data integration for gazetteer construction has been addressed by Hastings [3], though not in the context of a distributed access environment. His conflation methods employ geotaxical and geonomical semantic similarity metrics. Gazetiki constructs a gazetteer that integrates geographic concepts found on Wikipedia pages with the location derived from Panoramio photos for several European cities [10].

3 Overview of Toponym Ontology Model and Mediation System

We introduce a toponym ontology (TO), which is equivalent to what others may call a gazetteer, in combination with web service and data mediation functionality that enables access to multiple resources in response to a query on the TO. The main purpose of the TO is to support geo-parsing and reverse-geocoding (see section 1). Thus the TO has to 1) find toponym-data that matches a given input string representing a place name and 2) retrieve georeferenced place names given a spatial footprint as an input. For task 1 accurate coordinate data and wide geographic coverage are required from the gazetteer resource while for task 2 rich hierarchical information is required to provide unambiguous multi-part (hierarchical) toponyms.

The TO model, illustrated in Fig. 1, is based on the concept of a geofeature that corresponds to a named, spatially focused, geographical phenomenon, and conforms to Goodchild and Hill's definition of a place [1]. As with many gazetteers these minimum requirements are supplemented by other components of information. In particular this includes data about the source of the toponym data, the language of the name itself and of its alternative names, and hierarchical links to parent geofeatures of which the current geofeature is a part. The footprint may take the form of a point, a line, a simple polygon or a minimum bounding box, as well as a collection of one or more of these simple types. The footprint is also associated with the definition of the spatial reference system and a datum if available. The feature types are taken from a concept ontology developed in parallel with the TO and which includes scene types that were developed to support the specific applications of the TO.

The TO is to a large extent a virtual store of geofeatures. A query to the TO results in retrieval of TO data from remote geo-data sources, which are integrated on the fly.

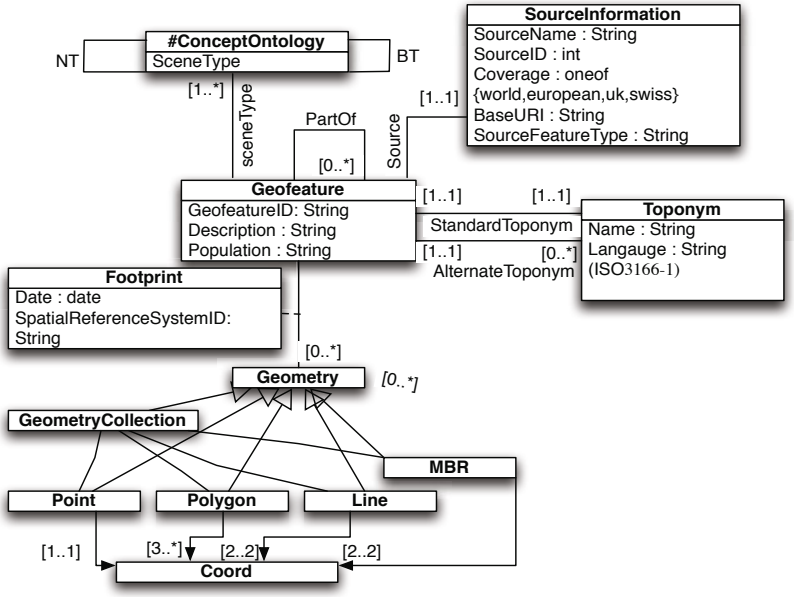


Fig. 1. Toponym Ontology Model

Because some of the toponym resources are not supported by web service access, the remote access procedures are supplemented by local database storage of these resources. For reasons of efficiency the TO maintains a local cache of the results of remote access calls and of the results of the integration and augmentation procedures. The framework for remote access is based on the mediation architecture introduced by Wiederhold [11]. It resembles the distributed retrieval engine approach of Callan [12] in being able to access, format and integrate local or remote geo-data sources on demand.

A three layer mediation architecture [11, 13] is employed consisting of a Foundation layer, Mediation layer and Application layer (see Fig. 2). The resource manager acts as an intelligent mediator handling on the fly access to the multiple heterogeneous toponym resources. The remote sources (e.g. Geonames[1]) are connected to and queried via their web service endpoints (Foundation Layer). Local resources are maintained in spatial databases and queries are issued using database connectors. The principal components of the resource manager are the interfaces to each resource (termed *Interface* in Fig.2) , the *Geofeature Integration Module* (GIM) and the *Geofeature Augmenter*. We summarise the characteristics of resource interfaces in this section, while deferring explanation of the GIM and Geofeature Augmenter until section 5.

Each of the accessible data resources employs its own data schema and output data formats, while each of the remote sources also employs its own set of web services

[1] http://www.geonames.org/

adapted to their respective schema. To deal with this the Resource Engine implements a separate interface to each resource. These interfaces query the resource's end point (local or remote) and formats results according to the internal geofeature model of the Toponym Ontology. The Resource Interface is comprised of two components, the *query translator* and the *results translator*.

In view of the divergent data schema and web service interfaces, the interface and internal data model of the toponym ontology can be both logically and syntactically incompatible with each resource [14]. The purpose of the *query translator* is therefore to morph standardised geofeature queries, issued to the Toponym Ontology engine, to a form that correctly queries each resource (where the schema of each resource can easily be interrogated [15]), and hence returns the information required to instantiate one or more geofeatures. Consequently, the resource manager is a fat mediator, in that all processing of source information is performed internally, and is not delegated to each source [16].

The purpose of the *results translator* is to convert between the native result format of the resource and the Toponym Ontology geofeature model. Translation is important in order to maintain a uniform internal data model, and is defined manually per resource as part of the resource interfaces (Fig. 2).

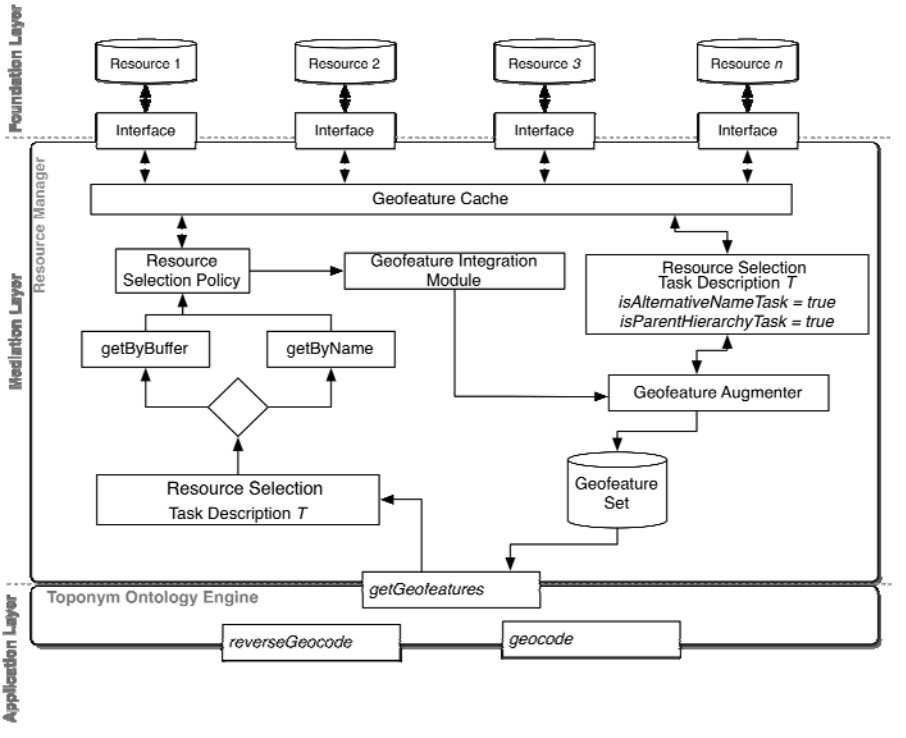


Fig. 2. The mediation architecture

4 Analysis of Data Resources Accessed by the Toponym Ontology

4.1 Data Sources

At present the remote sources employed are Geonames gazetteer, OpenStreetMap[2] (OSM), Yahoo Where on Earth[3] and Wikipedia[4] georeferenced articles, all subject to Creative Commons Licences. These are either accessed on demand or periodically downloaded in bulk. Local stored data, with no web access methods, include the Ordnance Survey 1:50,000 scale gazetteer (OS50K) and parts of their PointX and Mastermap products. Table 1 summarizes the characteristics of the resources.

Yahoo Where on Earth (YahooWOE), attempts to provide a permanent unique identifier (a WOEID) for every place on the Earth's surface. The standard API does not provide any reverse geocoding functions. Instead, reverse geocoding functions are obtained from Flickr, which has an extended API that wraps the existing YahooWOE API.

Table 1. Resources of typed geofeatures used in the topoym ontology : (H) indicates that a hierarchy is supported

Source	*# Features*	*Cov.*	*Geometry*	*Access*	*Format*
Geonames	*7 Mio. (H)*	*World*	*Point*	*Remote / Local*	*XML*
OSM	*Unknown*	*World*	*Point/ Polygon*	*Local*	*ESRI shape*
YahooWOE	*Unknown (H)*	*World*	*Point*	*Remote*	*XML*
Wikipedia	*~ 12. Mio*	*World*	*Point*	*Local*	*XML, RDF*
OS Point X	*3.9 Mio*	*GB*	*Point*	*Local*	*GML*
OS 50K	*~ 260k*	*GB*	*Point*	*Local*	*Ascii*
OS MasterMap	*> 10 Mio.*	*GB*	*Point/Polygon*	*Local*	*GML*

Many of the 12 million plus articles in the online multi-lingual collaborative encyclopaedia Wikipedia, have geographic content, being georeferenced by a latitude and longitude, and including alternative names. We imported a database dump of georeferenced articles into the TO. The remaining three Ordnance Survey data sets in table 1 are commercial administrative-oriented products.

4.2 Comparison of Sources

Here we compare locally stored version of four resources – Geonames, Wikipedia, OSM and OS50K - with regard to spatial distribution and some aspects of their associated attribute data. YahooWOE is excluded, due to being remote access only, as are Mastermap and PointX due to limited availability in our project. Spatial distribution is analysed using quadrat counts [17] that divide an area into rectangular sub regions of equal size. The number of toponyms whose footprint intersects each of these quadrats is then counted and recorded for each quadrat (see Fig. 3 and Table 2).

With regard to spatial distribution of toponyms the OS50K has the most uniform distribution across urban and rural areas. It has also the highest average number of toponyms per quadrat and the highest total number of toponyms. For certain areas

[2] http://www.openstreetmap.org/
[3] http://developer.yahoo.com/geo/
[4] http://www.wikipedia.org/

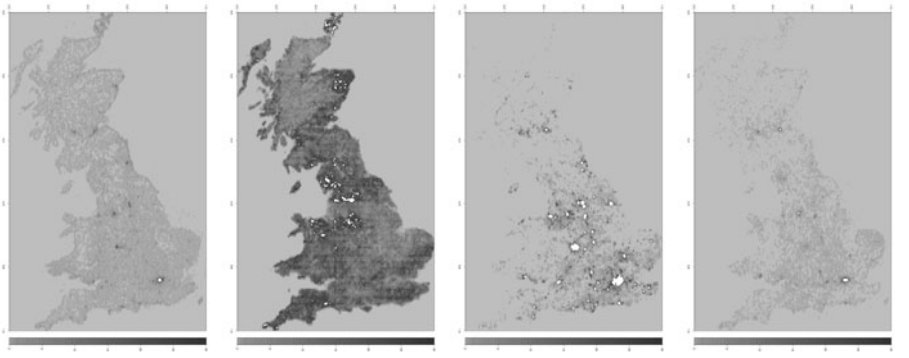


Fig. 3. 2D filled contour plot of the quadrat count for each of the four data sources in Great Britain, where each quadrat is 3.3km wide and 5.6km high. The contour plot for each source has the same range of z values and hence colour gradient. Values above 30 are coloured white. (a) Geonames, (b) OS50k, (c) OSM and (d) Wikipedia.

Table 2. Number of toponyms per web resource

	Geonames	*Wikipedia*	*OSM*	*OS50K*
Total Toponyms	31674	13030	49119	255182
Avg. Toponyms per Quad	0.79185	0.32575	1.227	6.379
Max Toponyms per Quad	245	300	998	58

however each of the other sources has higher local maximum numbers of toponyms per quadrat than OS50K, particularly within cities. For the UK, Geonames has the second best uniformity of spatial distribution with very good uniformity of distribution for Europe as a whole. OSM is notable for a bias towards to urban areas.

With regard to types of features in the resources, OSM, has relatively few larger scale features such as towns, cities, mountains and countries, but it records many buildings and some landmarks. However there is a bias to commercial service locations such as places of entertainment, ATM machines, garages and restaurants. This emphasis upon small scale features makes it applicable to reverse-geocoding applications for finding small scale features, such as within a photograph's viewport, as opposed to geocoding applications which require data sources rich in prominent landmarks along with larger scale features such as neighbourhoods, towns, cities and countries.

OS50K has many settlements and a range of other types of features but their feature typing suffers from a very large number of "other" unclassified features (about 50% for example in the city of Edinburgh and more in some rural areas). It is also limited by the low resolution of its coordinates (+/- 500m). Geonames is particularly rich in town/city scale features as well as natural geographic features such as rivers, lakes, mountains, coasts and valleys. Wikipedia is notable for having the highest numbers of well-known landmarks, which makes it good for reverse geocoding applications. Settlement classification for Wikipedia has limitations – for example many relatively small settlements are classed as "city". Both Geonames and YahooWOE have consistent, but different, parent hierarchies.

Geonames hierarchies are administrative and can only be determined for Geonames toponyms, while YahooWOE provides smaller scale neighbourhood (sometimes vernacular) levels up to country level parents with Global coverage. It should also be pointed out that Geonames and Wikipedia are good sources of alternative names which are useful for recognizing places for geocoding purposes. Geonames provides both language variations, alternative spellings and, on occasion, vernacular names.

5 Data Matching and Augmentation Procedures

Here we describe the functions of the resource manager for registering resources, and for matching, integrating and augmenting geofeatures from those resources. The resource manager registers all accessible resources, by adding entries into the resource register, including the name, ID, spatial coverage, data license, the uniquely constructed resource interface and the suitability of each resource. The suitability is encoded as Boolean flags associated with each of a set of tasks for which the Toponym Ontology may be employed. These include geocoding, reverse geocoding, toponym hierarchy construction and alternative names retrieval (see Table 3).

Table 3. Suitability of toponym resources for tasks

	Geonames	*Wikipedia*	*OS50k / OSM / Point X*	*YahooWOE*
Geocoding	√	√	√/-/-	-
Rev. geocoding	√	√	-/√/√	-
Hierarchy	√	-	-	√
Alternative names	√	√	-	-

5.1 Geofeature Integration Module (GIM)

The Geofeature Integration Module (GIM) matches equivalent geofeatures from heterogeneous data sources. Equality is measured on the standard name and spatial location, using a spatiotextual similarity measure. The GIM operates over all sets of Geofeatures G (where $G = \{g_1,\ldots,g_n\}$, and g is an individual geofeature) returned from each source after a get geofeatures request has been issued.

Textual similarity between the standard names is determined using a combination of the Levenshtein metric to measure edit-distance [18], text normalisation (using ICU4J decomposition) and the SoundEx phonetic algorithm [19]. Text normalization uses IBMs International Components for Unicode (ICU) Java library (ICU4J[5]), which transforms composite characters into pre-composed characters. For example Zürich becomes Zu{dieresis}rich, where the diacritic mark {dieresis} has been removed from the character glyph (u with an umlaut). Removal of all decomposed diacritical Unicode characters results in a canonical form i.e. Zurich. The Levenshtein metric $Sim_{lvd}(w_1,w_2)$ measures the edit-distance between two strings w_1 and w_2, which is the number of edits (alterations such as copy, delete, insert, substitute) needed to change one string to another. Each type of edit is assigned a weighting. If the edit-distance is

[5] http://www.icu-project.org/

> 3 its score is set to 0 (edit distances > 3 indicates the two strings are too dissimilar to be considered), otherwise the score is computed as:

$$Sim_{lvd}(w_1, w_2) = \begin{cases} 0 & \text{if } lvd(w_1, w_2) > 3 \\ 1 - \dfrac{lvd(w_1, w_2)}{3} & \text{otherwise} \end{cases} \quad (1)$$

where, $lvd(w_1,w_2)$ computes the edit distance between two strings, and the final similarity measure is in [0,1], where 1 represents equivalent strings.

The SoundEx algorithm matches phonetically similar sounding words, using language dependent rules that allocate numerical values to phonetically distinct character groups. The SoundEx similarity measure computes the difference between two word strings w_1 and w_2. $sdx(w_1,w_2)$ will be a score from 0-4 where 0 represents no similarity and 4 indicates the strings are identical. The final measure $Sim_{sdx}(w_1,w_2)$ is a value in [0,1] where 1 denotes two strings are identical:

$$Sim_{sdx}(w_1, w_2) = \frac{sdx(w_1, w_2)}{4} \quad (2)$$

The combined edit distance and Soundex distance measure (denoted *sim*) is:

$$sim(w_1, w_2) = \frac{(4 * Sim_{sdx}(w_1, w_2) + 1 * Sim_{lvd}(w_1, w_2))}{5} \quad (3)$$

The weighting here is the result of parameter tuning during empirical testing, and reflects the higher confidence we have in using Soundex to match misspelt words. Note that toponyms are always normalised before $sim(w_1,w_2)$ is calculated.

The distance in metres between each pair of geofeatures $<g_1,g_2>$ that have a standard name similarity score > 0.8 is calculated using the geodesic arc distance [20] based on coordinates in the WGS84 spheroid coordinate system that is used for each source (following transformation from their original coordinate system as necessary). Each name-matched pair of geofeatures $<g_1,g_2>$ is then treated as a match if d is less than some value *min* (currently set at 50 metres).

The resulting set G may contain many matched pairs and, for each pair, only one geofeature; g_1 or g_2 is returned. Which to remove depends on the priority of the source of g_1 compared to the priority of the source of g_2. The source with the lower priority is removed from the pair. The list (manually created) of source priorities starting with the highest is: Wikipedia, PointX, Mastermap, Geonames, OSM, OS50K. The removal of matching geofeatures is an iterative procedure in which a lower matching priority geofeature will be removed from all pairs in which it occurs.

5.2 Resource Selection

The Resource Manager has two types of resource selection policies: 1) *Priority Selection* queries each source in a defined order until the query can be satisfied. 2) *Maximum Selection* queries each source in turn and filters the results, using the toponym matching procedure described above. The selected representation may then be augmented with data from other representations of the same place in a later processing stage described in the next section. *Priority Selection* is used by the geocoding function of the toponym ontology for which a single geofeature match is appropriate.

Reverse geocoding uses the *Maximum Selection* policy to obtain as many geofeatures as possible within a given spatial buffer. The function *isSuitableFor* takes as input the current resource *s,* and an input task description *T*, and returns true if the resource is suitable for the current task. A task description *T* indicates whether the task is geocoding, reverse geocoding, hierarchy retrieval, alternative names retrieval, or it can specify a particular source.

Algorithms for the two policies are presented in pseudo-code below where $Q(s_i)$ performs a query consisting of either *getByBuffer* (which returns all features within a given buffer) or *getByName* (which returns a set of geofeatures based on a fuzzy standard and alternative name match) on the given resource s_i. Source priority is currently, starting with highest priority: Geonames, OS50K, Wikipedia, Mastermap, OSM and PointX. The procedure *GIM(G)* employs the duplicate detection method described in the previous section. The resource manager can accept other types of query on the registered resources, e.g. relating to retrieval of hierarchical levels.

Algorithm: Priority Selection	**Algorithm: Maximum Selection**
Input: a geofeature query *Q,* a task description *T;* **Output:** a collection of geofeatures *G,* or an empty set if none was found; Let *S* be the set of registered sources; Order *S* by source priority; For each source s_i in *S* If *isSuitableFor*(s_i, *T)* Let $gc = Q(s_i)$; If $gc \neq \theta$ Return *gc;* Return. θ	**Input:** a geofeature query *Q,* a task description *T;* **Output:** a collection of geofeatures *G;* Let *S* be the set of sources in the resource Registry; Let *G* be an empty set of geofeatures; For each source s_i in *S* If *isSuitableFor*(s_i, *T)* Let $gc = Q(s_i)$; Let $G = G \cup gc$ Remove duplicates $G = GIM(G)$ Return *G*

5.3 Geofeature Augmentation

The resource manager can not only retrieve but also construct augmented geofeatures. Information from a number of sources is merged, on the fly during each request, to create more complete and consistent instantiations of geofeatures. The *Geofeature Augmenter (GA)* performs 1) *addition* of a consistent and accurate set of administrative parents to small scale geofeatures, e.g. POI; and 2) *reconstruction* of full and consistent geofeature records given an arbitrary geofeature.

We present a procedure for administrative hierarchy augmentation. It uses the YahooWOE hierarchy data to augment geofeatures retrieved from data sources such as OSM which have no explicit parent hierarchy, or OS50K and Wikipedia which only contain county or country level parents.

For administrative regions the algorithm only assigns a country level parent, as they already represent part of the administrative hierarchy. Large scale geofeatures such as lakes, parks and others are only given country level parent information, as they could span a number of administrative areas.

Algorithm: Administrative Parent Hierarchy Augmentation
Input: A geofeature *g*
Output: The same geofeature *g* with enhanced parent hierarchy from YahooWOE
Create task description *T*, set *sourceName* = YahooWOE
Let *P* be the set of parents returned by querying getParentHierarchy(*g.location*,*T*)
Attach the parent hierarchy *P* to *g*
Return *g*

The algorithm *Geofeature Reconstruction* presented below takes as input a geofeature *g* and outputs an augmented version of the same geofeature, following a process of replacement or addition of attributes from matching geofeatures. In the procedure documented here the task description simply specifies a single resource of Geonames that is to be used for matching against the input, as this resource is known to be good quality with regard for example to alternative names, population statistics and feature type. It is possible to extend this procedure to augment from multiple sources. The reconstruction procedure uses a function *STEquiv* which takes as input a set of geofeatures retrieved by a query to the resource and matches them to the single input geofeature, using the GIM matching methods. If an equivalent Geonames geofeature is not found, *g* is returned unchanged.

Algorithm: Geofeature Reconstruction
Input: A geofeature *g*
Output: The same geofeature *g* with enhanced attributes
Create task description *T*, set *sourceName* = Geonames
Let *G* be the set of geofeatures returned by querying
getByName(*g.standardName*,*T*)
Let *ge* = *STEquiv (g,G)*
If *ge* ≠ null
Set null values of *g* to those of *ge*
Return *g*

As an example of geofeature reconstruction, consider the following. YahooWOE is a good resource for finding parent containment for point locations through the Flickr API[6]. However YahooWOE is often limited with respect to the number and types of attributes returned. Fig. 4 illustrates augmentation for the geofeature Cardiff, which was retrieved from YahooWOE as the 'Region' parent of the location 51.47, -3.19.

[6] http://www.flickr.com

The place record retrieved from YahooWOE for Cardiff is typical in having no alternative names, no population information and only a rather broad type classification i.e. 'Region'. Consequently, this record is augmented with alternative names, population information and a more appropriate place type from a matching Geonames entry.

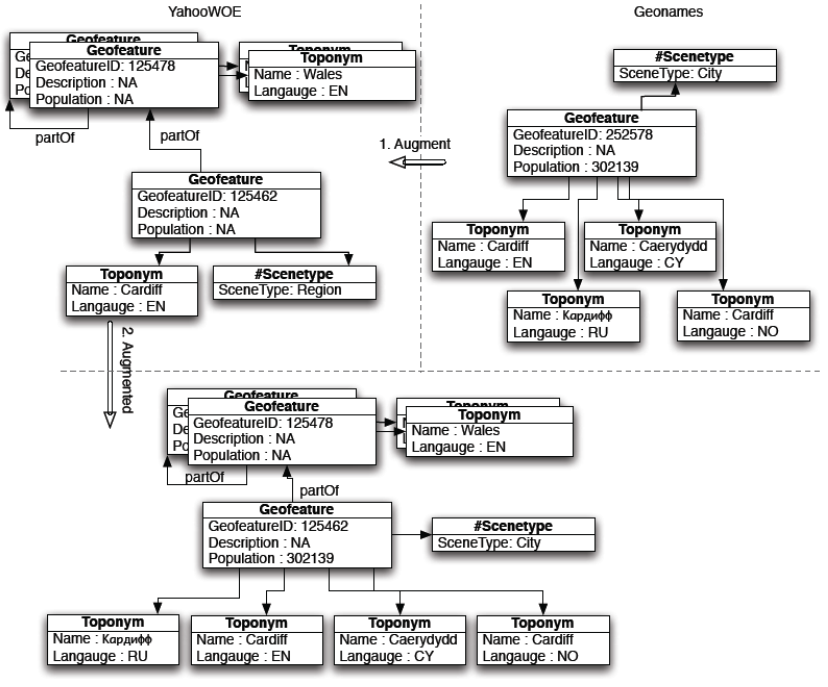


Fig. 4. Example reconstruction of the Cardiff Yahoo WOE geofeature with the Cardiff Geonames geofeature

6 Results and Evaluation

To test the accuracy of the Geofeature Integration Module (GIM), five different locations were sent to the *getByBuffer* method of the resource manager (each with a 200m buffer), and comparisons from the GIM over the set of returned geofeatures were manually examined. Table 4 shows each of the five locations, the number of geofeatures returned from the resource manager, the number of manually identified matching pairs, the number of matching pairs which were successfully matched and resolved (where one is then removed), the number of similar pairs which were not matched (failures), and those that where matched but should not have been (false positives).

The results show the *GIM* to be successful in removing 67.50% of matching names between sources. It is also notable that, for this sample set, the spatio-textual measure does not produce any false positives, i.e. there are no generated matches which are known, by manual investigation, to be incorrect.

Table 4. Evaluation of the *getByBuffer* method

Location	*Returned Geofeatures*	*Similar Pairs*	*Success*	*Failures*	*False Positives*
1 – Cardiff	5	2	*1 (50%)*	*1 (50%)*	0
2 – Edinburgh	5	2	1 (50%)	*1 (50%)*	0
3 – Cardiff	2	1	1 (100%)	0 (0%)	0
4 – Edinburgh	*9*	3	3 (100%)	0 (0%)	*0*
5 – Cardiff	*25*	8	3 (37.5%)	5 (62.5%)	*0*
Average			67.5%	32.5 %	0%

Observations of successful matches include: high accuracy in matching locations which start with similar word-grams and have similar locations e.g. 'Cardiff Arms Park (Cardiff RFC)' (OSM) and 'Cardiff Arms Park' (Wikipedia) which have a textual similarity score of 0.85 and a distance of 41m; identical name matches between sources e.g. 'Royal Botanic Garden Edinburgh' in both Wikipedia and OSM with a 48m difference between locations; and subtle differences in punctuation e.g. 'St James Centre' in OSM compared to 'St. James Centre' from Wikipedia.

Observations of failed matches include: one source having duplicate entries with initial word-gram name variations that give low Soundex similarity scores e.g. 'Cardiff Millennium Stadium' (Geonames) and 'Millennium Stadium' (Geonames); use of abbreviations in sources leading to large word variations and high edit-distance scores e.g. the user contributed entry in OSM 'Univ Liby' compared to its proper name 'University Library' from Mastermap; and locations with high textual similarity but separated by distances exceeding 50m, e.g. 'Pont Sticill' (Geonames) and 'Pontsticill' (OS Gazetter) with 0.933 name similarity but a 745m difference in location.

7 Conclusions

This paper has addressed the need to access heterogeneous gazetteer data available in the combination of volunteered and formal resources. Our mediation-based metagazetteer service supports integration methods that conflate multiple attributes from the different resources using a toponym feature matching procedure. The resource selection and priority strategies were based on a prior analysis of their data characteristics in combination with application requirements. Future work will present the results of already conducted application-oriented evaluations that demonstrate the effectiveness of the methods presented here in practice. It will also focus on automated methods to rank resources and their component data items as well as further refinement of the methods for toponym equivalence determination.

Acknowledgements

This research was supported by funding from the European Commission project TRIPOD (IST-FP6-045335) and from the UK Ordnance Survey.

References

1. Goodchild, M.F., Hill, L.L.: Introduction to Digital Gazetteer Research. International Journal of Geographic Information Science 22(10), 1039–1044 (2008)
2. Goldberg, D.W., Wilson, J.P., Knoblock, C.A.: Extracting Geographic Features from the Internet to Automatically Build Detailed Regional Gazetteers. International Journal of Geographical Information Science 23(1), 93–128 (2009)
3. Hastings, J.T.: Automated Conflation of Digital Gazetteer Data. International Journal of Geographical Information Science 22(10), 1109–1127 (2008)
4. Keßler, C., Janowicz, K., Bishr, M.: An Agenda for the Next Generation Gazetteer: Geographic Information Contribution and Retrieval. In: 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009, Seattle, Washington, USA (2009)
5. Mikheev, A., Moens, M., Grover, C.: Named Entity Recognition without Gazetteers. In: Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics, pp. 1–8 (1999)
6. Uryupina, O.: Semi-Supervised Learning of Geographical Gazetteers from the Internet. In: HLT-NAACL 2003, Workshop on Analysis of Geographic References, Alberta, Canada, pp. 18–25 (2003)
7. Christen, P., Churches, T.: A Probabilistic Deduplication, Record Linkage and Geocoding System. In: ARC Health Data Mining Workshop, Canberra, AU, The Australian National University (2005)
8. Hollenstein, L.: Capturing Vernacular Geography from Georeferenced Tags, Msc Thesis, in Department of Geography, University of Zurich (2008)
9. Rattenbury, T., Naaman, M.: Methods for Extracting Place Semantics from Flickr Tags. ACM Trans. Web 3(1), 1–30 (2009)
10. Popescu, A., Grefenstette, G., Moëllic, P.-A.: Gazetiki: Automatic Creation of a Geographical Gazetteer. In: JCDL 2008: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries, pp. 85–93. ACM, New York (2008)
11. Wiederhold, G.: Mediators in the Architecture of Future Information Systems, vol. 25(3), pp. 38–49. Computer IEEE Computer Society Press, Los Alamitos (1992)
12. Callan, J.: Distributed Information Retrieval. In: Advances in Information Retrieval, pp. 127–150. Kluwer Academic Publishers, Dordrecht (2000)
13. Smith, J.M., Bernstein, P.A., Dayal, U., Goodman, N., Landers, T., Lin, K.W.T., Wong, E.: Multibase - Integrating Heterogeneous Distributed Database Systems. In: AFIPS National Computer Conference (1981)
14. Gupta, A., Marciano, R., Zaslavsky, I., Baru, C.K.: Integrating GIS and Imagery through XML-Based Information Mediation. In: Agouris, P., Stefanidis, A. (eds.) ISD 1999. LNCS, vol. 1737, pp. 211–234. Springer, Heidelberg (1999)
15. Zaslavsky, I., Gupta, A., Marciano, R., Baru, C.: Xml-Based Spatial Data Mediation Infrastructure for Global Interoperability. In: 4th Global Spatial Data Infrastructure Conference (2000), `http://www.gsdi.org/capetown/program.htm`
16. Gupta, A., Memon, A., Tran, J., Bharadwaja, R.P., Zaslavsky, I.: Information Mediation across Heterogeneous Government Spatial Data Sources. In: Annual National Conference on Digital Government Research, pp. 1–6. Digital Government Society of North America, Los Angeles (2002)
17. Diggle, P.J., Besag, J., Gleaves, T.J.: Statistical Analysis of Spatial Point Patterns by Means of Distance Methods. Biometrics 32(3), 659–667 (1976)

18. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
19. Russell, R., Odell, M.: The Soundex Indexing System. National Archives and Records Administration (1918), http://www.nara.gov/genealogy/coding.html
20. Sinnott, R.W.: Virtues of the Haversine. Sky and Telescope 68(2), 159–162 (1984)

Qualitative Change to 3-Valued Regions

Matt Duckham[1], John Stell[2], Maria Vasardani[3], and Michael Worboys[3]

[1] Department of Geomatics, University of Melbourne, Victoria, 3010, Australia
matt@duckham.org
[2] School of Computing, University of Leeds, Leeds, LS2 9JT, UK
j.g.stell@leeds.ac.uk
[3] Department of Spatial Information Science and Engineering, University of Maine, Orono, ME, 04469, USA
{mvasardani,worboys}@spatial.maine.edu

Abstract. Regions which evolve over time are a significant aspect of many phenomena in the natural sciences and especially in geographic information science. Examples include areas in which a measured value (e.g. temperature, salinity, height, etc.) exceeds some threshold, as well as moving crowds of people or animals. There is already a well-developed theory of change to regions with crisp boundaries. In this paper we develop a formal model of change for more general 3-valued regions. We extend earlier work which used trees to represent the topological configuration of a system of crisp regions, by introducing trees with an additional node clustering operation. One significant application for the work is to the decentralized monitoring of changes to uncertain regions by wireless sensor networks. Decentralized operations required for monitoring qualitative changes to 3-valued regions are determined and the complexity of the resulting algorithms is discussed.

Keywords: qualitative spatial reasoning, uncertainty, topological change.

1 Introduction

Imagine a collection of sensors, situated in the plane, measuring values of a single scalar quantity, such as temperature, water salinity, or gas concentration. These measurements may be repeated over an extended time period, thus resulting in time-series at each local sensor node. The overall problem, from which this research stems, is how to construct some high-level qualitative global descriptions of dynamic field of values as it evolves through time.

In the restricted case where the measurand recorded by each sensor takes a Boolean value, we may objectify the dynamic field as a collection of regions of high (or low) intensity that is evolving over time. However, the restriction of the measurand to the Boolean domain provides only a first approximation to qualitative representations of field evolution. A richer approximation would allow the measurand to take one of n values, for some positive integer n. In this paper, we move to the case where the measurand domain is three-valued. This case can model several different kinds of scenarios:

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 249–263, 2010.

1. The useful model for the domain really is three valued, and may be ordered or unordered.
2. The useful model for the domain is Boolean, but some sensors cannot determine which of the two values applies at a particular moment. This may occur because of measurement uncertainty or conflict among the sensors, or the sensor is in hibernation mode or not working, or because the domain itself is inherently vague.

For instance, various wireless sensor networks (WSN) have been set for monitoring dynamic spatial fields such as temperature, humidity, and barometric pressure, the combination of which provide indications for potential wildfires [14, 2]. Due to the variability in combinations of the measurands, neighboring sensors may be providing mixed Boolean values. For example, in the network of Figure 1, sensors marked with black filled circles indicate fire, while the ones marked as clear circles indicate safe area. The highlighted area, however, includes readings of both cases and is an area of uncertainty, which cannot be identified either as region on fire or as safe region. The same uncertainty is met in cases of flood monitoring using water level sensors (e.g., ALERT [3]), or in the detection of toxic gas dissemination using air quality sensors.

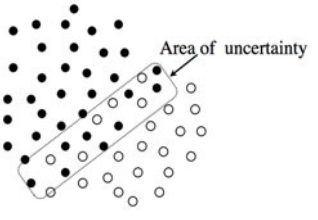


Fig. 1. An area of uncertainty due to mixed neighboring sensor readings

The need to abandon the strictly Boolean domain may also result from difficulties in the deployment of a sensor network due to the hostility or scale of the monitored area. In applications such as the joint efforts of Harvard University, the University of New Hampshire, and the University of North Carolina to deploy a WSN to monitor eruptions of Volcan Tungurahua, an active volcano in Ecuador [22] or NASA's VolcanoWeb [23], it may be impossible to cover certain hostile parts of the monitored environment. Sensor networks that serve the study of natural phenomena that intrinsically discourage human presence and spread over large areas, such as hurricanes and tsunamis, may suffer from low density and result in uncertainty in defining the geographic spread of the phenomenon [21].

In different applications, unreliable measurements may be the cause of uncertainty. When offshore wireless sensors are attached to floating buoys (e.g., the CORIE network along Columbia river), the direct light-of-sight is frequently obscured because the height of surface waves exceeds the height of the onshore antennas. The highly dynamic connectivity of such networks may result in uncertain and unreliable sensor readings. Likewise, uncertainty may result from hardware failure in the sensors, which causes coverage holes in the network [1, 21]. Yet a different case is that of responsive WSN—adaptive networks that track salient changes and only keep a certain number of sensors active, leaving the rest of the sensors in hibernation mode

[7]—where the dynamic field is monitored by evolution of surface regions that are better represented as fuzzy regions, due to the uncertainty in the extent of the measurand.

For such WSN applications, we may then consider the dynamic field as a three-valued surface, evolving through time, or as a dynamic planar map, where each face takes one of three values. In the latter case, particularly when we are dealing with semantics related to uncertainty, we can think of an evolving collection of regions with broad boundary, which is the approach considered in this work. One method of detecting the topological changes in such regions with broad boundaries would be to communicate data from all nodes to a centralized server, and then process this information using standard tools, like a GIS or a spatial database. However, such centralized approaches are acknowledged to be inefficient, unscalable, and subject to bottlenecks and a single point of failure. The limited bandwidth and node energy resources in untethered networks demand *decentralized* algorithms that are able to detect salient changes in the network itself. In a decentralized algorithm, no single node ever possesses global knowledge of the system state [15]. Therefore, the method presented in this paper is based on a decentralized algorithm approach.

The remainder of this paper is structured as follows: Section 2 provides an outline of the existing literature on detecting changes in crisp or broad boundary regions. In Section 3 we provide the theoretical framework for qualitative change representation, introducing a new tree structure, namely the *BB containment tree*, which allows the depiction of topological events that occur to broad boundary regions. Section 4 provides a decentralized algorithmic approach for monitoring such changes, and Section 5 makes some conclusions.

2 Related Work

This section summarizes the background research on which the work in this paper is based, as well as some related work on the subject of moving vague regions. Previous work, by the authors and others [25, 12, 13] has discussed the case where the measurand is Boolean, providing a formalization and classification of topological changes to regions, as well as a collection of algorithms for topological change detection in a decentralized setting, such as a wireless sensor network. We assume that the regions are bounded by non-intersecting closed curves, embedded in the Euclidean plane; of course, this is a generalization that favors convenience, since the application domains considered are mostly continuous. In this case, a collection of regions in the plane is represented as a rooted, directed tree, where the nodes of the tree are the regions. The root of the tree, marked with a double circled node, is the unbounded outer region, and a directed edge exists between two nodes if they share a common boundary. An example is shown in Figure 2.

Topological change to the region collection can be defined by specifying kinds of relations that exist between their corresponding trees. In particular, this formalism can be used to characterize "atomic changes" to the region collection: region insertions and deletions, as well as merges (two kinds—normal and self-merge) and splits (two kinds—normal and self-merge). Geometric properties of region collection evolution may then be represented algebraically, and in some cases this makes the analysis simpler. For example, it is possible to show that there is a normal form for any complex change of a collection of regions as a composition in a specific sequence of atomic changes [13].

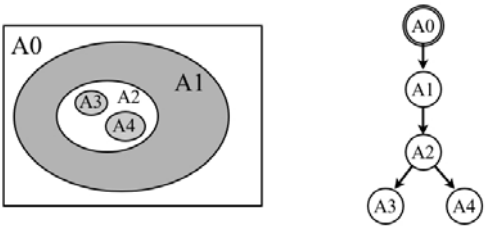


Fig. 2. A collection of regions in the plane and their tree representation

While previous research focused on the analysis of types of topological changes that can occur to planar regions through time as they are extracted from the Boolean domain, in this paper focus is on description of the evolution of regions with broad boundaries as they emerge from 3-valued dynamic fields. Such vague regions whose boundaries cannot be easily determined are represented by an egg-yolk pair [4], which is made of two concentric regions. The inner region, referred to as the yolk, represents the parts that definitely belong to the region, while the surrounding region, referred to as the white represents the parts that may or may not belong to the region.

Ibrahim and Tawfik [9, 10] have extended the egg-yolk theory from dealing with spatial regions only, to dealing with regions of space-time. This approach is based on Muller's [17] theory of spatiotemporal regions, which share spatiotemporal, as well as temporal relations. RCC-8 is chosen as the spatial theory combined with Allen's temporal relations and seven motion classes *mc* are defined: *leave, reach, cross, hit, internal, external,* and *split* [17]. Ibrahim and Tawfik [9] redefine the motion classes for vague (egg-yolk) regions. This approach is different than the method of following the evolution of broad boundary regions as the regions themselves are essentially one-dimensional and can only undergo a limited range of qualitative change.

3 Modeling Regional Change

Starting with a potentially uncertain property of locations, we can divide the plane into regions of three types. These are: (1) the regions which definitely have the property, (2) the regions which may or may not have the property, and (3) those regions that definitely do not have the property. The regions of the first two types together may be seen as making up regions of a more complex kind—the broad boundary regions. In order to determine the most appropriate formal structure to model changes to broad boundary regions, we first need to examine the more general question of how we might model change to crisp regions of three types in the plane. By doing this without initially assuming that some of these regions taken together constitute broad boundary regions, we are able to see more clearly how the formal structure generalizes that already used for crisp regions of just two types.

In modeling change it is not sufficient to deal only with starting and ending configurations. We also need to know which regions at the start are related to which regions at the end, and also how this relationship was brought about (by splitting, merging, etc.). To do this with 3-valued regions, it is necessary to extend the tree representation illustrated in the two-type case in Figure 2 (Section 2), and to develop the appropriate notion of relation between these extended trees.

3.1 Varieties of Tripartite Division

We start with a three-way, or tripartite, division of the plane into regions which have boundaries consisting of non-intersecting closed curves. The most general type of such a tripartite division places no limitation on what types of region may appear within each other. This might be interpreted as having three types of substance which do not mix with each other. The dynamics of this situation are then a straightforward extension of the two-type case. Consider one aspect: the splitting of a region. Region splitting may be divided into splits and self-merges. These are distinguished topologically in the plane, as in a split one boundary divides into two boundaries neither of which contains the other, but in a self-merge one boundary divides into two one of which lies inside the other. In the three-region analysis we get six possibilities. This simply provides a combinatorial multiplication of the cases with no structurally significant new features being introduced.

3.1.1 The Intermediate Case

In some applications it is appropriate to regard one type of region as intermediate between the other two. Here it is natural to use black, grey, and white colors in diagrams of the regions, with grey as intermediate between the other two. To ensure grey regions are intermediate means that we require that between any white area and any black one there must be a grey region. This approach might be used to model height of a surface as indicated in Figure 3.

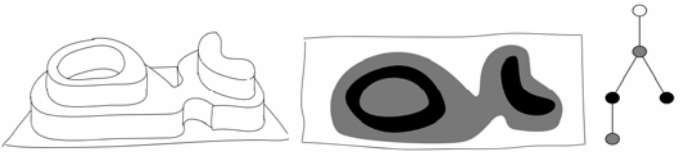


Fig. 3. A schematic landscape, its corresponding height regions and tree

With this model we can capture notions such as saddle points and plateaus containing both higher and lower areas. Changes to landscape, such as erosion of higher regions to produce level areas and the formation of valleys separating between two relatively higher areas, can be described.

3.1.2 The Broad Boundary Case

A particular case of three types of region is provided by regions of uncertainty. In this case the grey areas are indeterminate regions and the black and white areas are known regions. This type of indeterminate region is the region with a broad boundary, consisting of a certain core and an uncertain area surrounding it. Once we consider the dynamic behavior of such regions we see that it is impossible to be restricted to grey areas which contain a single known (black or white) part. If three such regions merge we can meet cases such as in Figure 4.

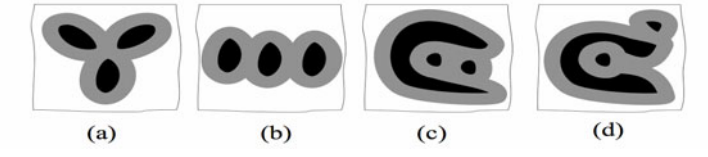


Fig. 4. Topologically equivalent situations within a broad boundary

The examples shown in Figure 4 are all indistinguishable from a purely topological viewpoint. Each consists of one grey region containing three black regions within it. Informal descriptions of these examples can of course easily distinguish between them. It is possible to see Figure 4(a) as three broad boundary regions which have their broad boundaries fused at one place. In example (b) we can see three broad boundary regions, but this time one is in between the other two. In (c) it is less clear that there are three broad boundary regions, but there is a sense in which the two smaller black regions are inside the larger one. This is not being topologically inside, but is inside in the sense of being inside the convex hull. Finally in (d), only one of the small regions is visually enclosed by the large one, and the other small crisp region is outside.

3.2 Containment Sensitivity

Considering Figure 4(d) we may wish to distinguish three separate regions, but with only one region of uncertainty containing the broad boundaries of all three regions (Fig. 5). This means that in the tree, a grey node no longer stands for a particular region but for a notional broad boundary. To represent this in our tree we need to group these grey nodes together.

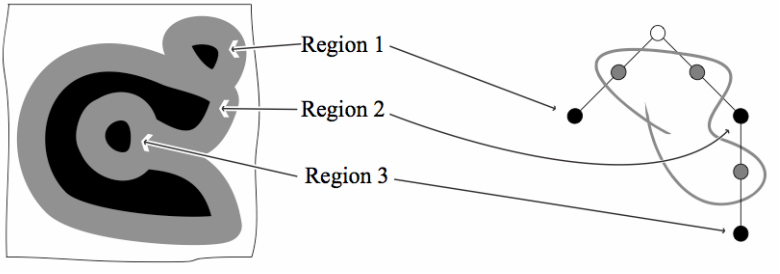


Fig. 5. Notional separation into regions and associated tree

The next step is a precise description of the kind of clustering the grey nodes shown in Figure 5. Suppose we have a set X which carries a symmetric relation φ. A subset $A \subseteq X$ is said to be φ-connected if given any $a, b \in A$, there is a sequence a_0, a_1, …, a_n of elements of A such that $a = a_0$, $b = a_n$, and $a_i \varphi a_{i+1}$ for $i = 0,\ldots, n$ -1. We apply this in the case that φ is the symmetric closure of the adjacency relation in a directed tree. We assume that the three colors of nodes in a tree are grey, black and white; the last two colors used for the 'certain' nodes and the first color for the 'uncertain' ones.

Definition 1. *A **BB containment tree** is a 3-coloured directed tree (T, α) with the constraint that a grey node can be incident with exactly two crisp nodes and where there is an equivalence relation on the set of all grey nodes which is α-connected.*

Given an arrangement of regions in the plane we construct a BB containment tree as follows. Start by constructing a tree of the crisp nodes only. The children of a node k are those nodes n representing regions which lie in the convex hull of k and for which

there is no crisp region between n and k. Then augment this tree by adding between every two crisp nodes a grey node and write $m \; \alpha \, n$ if there is an edge from m to n in the resulting tree. Now impose a relation R on grey nodes, making $g \, R \, g'$ iff there is a node n such that $n \; \alpha \, g'$ and also either $g \; \alpha \, n$ or $n \; \alpha \, g$. Finally make two grey nodes equivalent if the symmetric and transitive closure of R makes them so.

We demonstrate this construction by the example shown in Figure 6. The equivalence classes of grey nodes are indicated by clustering them together in the diagram. The example demonstrates that the structure captured by the tree enables us to distinguish between a region being topologically contained within another and only being in the convex hull of another. A node m represents a region topologically contained in the region represented by n if the grey node between m and n is not to equivalent to the grey node above n.

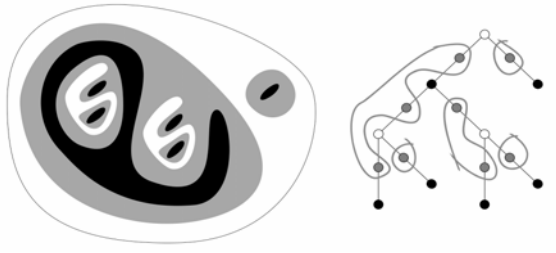


Fig. 6. Regions and the associated BB containment tree

3.3 Dynamics

Now we consider how to model the dynamic behavior to the broad boundary regions an example of which is provided by Figure 7.

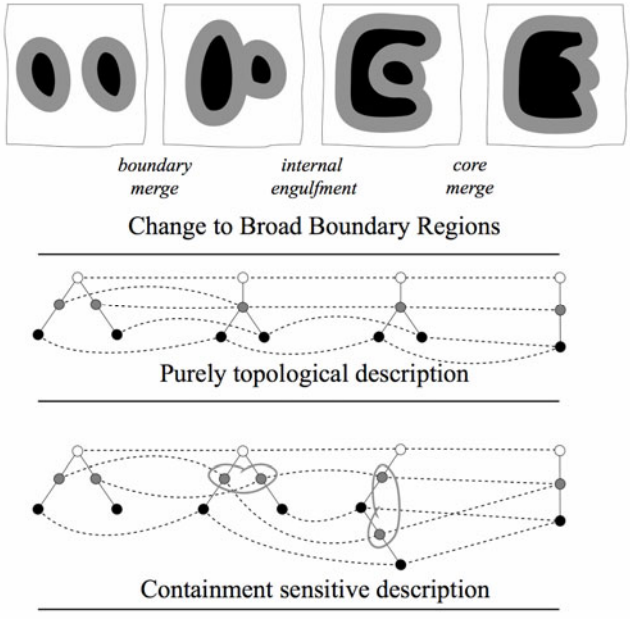


Fig. 7. Example of broad boundary region dynamics

Before two BB regions merge into a BB region with a merged core there must be a *boundary merge* as illustrated in the figure. If we model only the topological relationships between the black and grey regions we are unable to detect the stage described as *internal engulfment* in the diagram. This change is modeled by a new type of tree modification in which two certain nodes adjacent to equivalent grey nodes and at the same depth in the tree may move so that one is below the other, while maintaining the equivalence relation on grey nodes.

3.4 Contraction to Topology

Although we have seen that the BB containment tree provides a more detailed account of the spatial relationships than is obtained from topology alone, it is important to have a formal justification of the relationship between these two models. One reason is that it is sometimes more appropriate to use the purely topological model and thus being able to derive this from the containment sensitive model avoids the need to maintain two separate structures. Another reason is that by showing how sequences of successive containment sensitive configurations are mapped on to sequences of configurations described only topologically allows us to justify that the new technique is an extension of the existing one. The idea is based on contracting the tree, as shown in Figure 8.

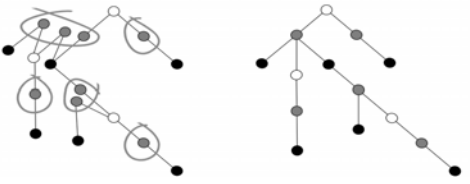


Fig. 8. Contracting the BB containment tree to its topological equivalent

We form the collapsed tree (T/R, β) as follows. The nodes of T/R are the crisp nodes of T together with the equivalence classes of grey nodes under the relation R. The tree is intermediate—one grey node between any two crisp nodes—so we specify when a crisp node k is adjacent to a gray node G and vice versa there being no other edges (1, 2). In the definition we use α^+ to mean the symmetric transitive closure of α.

$$k \ \beta \ [g] \ \text{iff} \ k \ \alpha \ [g] \ \text{and} \ \neg\exists h \in [g] \ (h \ \alpha^+ \ k) \tag{1}$$

$$[g] \ \beta \ k \ \text{iff} \ g \ \alpha k \tag{2}$$

It does need to be justified that T/R is actually a tree and not merely a graph. To see that this is the case it is sufficient, because of the intermediate nature of the graph, to show that between any two crisp nodes in T/R there is a unique path. This can be done by considering operations of expansion and contraction for paths between crisp nodes. The idea is that any path between crisp nodes in T can be contracted to a path between the same nodes in T/R and any path between the nodes in T/R can be expanded to a path in T.

To explain expansion first, any path in *T/R* will be made up of triples of the form $k\ G\ k'$ where k and k' are crisp and G is a grey node. The triple $k\ G\ k'$ is expanded to the unique path in T between k and k', and by expanding all triples we expand the whole path. Contraction is a reverse process in which the conditions on the equivalence relation ensure that arbitrary paths can be split up into sections which can be individually contracted to triples of this form.

4 Decentralized Algorithm

Previous work has already established a number of important algorithmic constructs that can be used as the basis of an extended algorithm for detecting topological changes in regions with broad boundaries. Specifically, two key classes of decentralized algorithms have already been investigated:

1. Decentralized algorithms for boundary detection and topological change in (crisp) regions [8, 19, 25]; and
2. Decentralized algorithms for detecting the structure of complex areal objects, including (crisp) regions with holes and islands [6, 20].

The algorithms (described in more detail below) construct decentralized analogs of fundamental centralized data structures and operations: specifically polygonal data structures augmented with boundary orientation (for example, as defined in ISO 19107, [11]), and the semi-line algorithm for point in polygon tests [24]. However, unlike their centralized counterparts, in these decentralized algorithms no single node has access to the global system state. Instead, information is generated, processed, and stored throughout the network itself, in particular at the boundary of the regions themselves.

4.1 Detecting Topological Changes in Crisp Regions

Detecting topological changes in crisp regions essentially involves two stages. First, each individual node detects if they are the boundary of a region by querying their immediate one-hop neighbors. Nodes that detect the region using their sensors (e.g., high temperature, low salinity) but are adjacent to nodes that do not detect the phenomenon are adjudged to be at the (crisp) boundary [25].

Second, a higher-level boundary structure is constructed to mediate collaboration across boundary nodes in detecting topological change. For example, [8] and [19] describe algorithms to construct cycles of nodes around the boundary of the region, rather like decentralized "polygons." Both these algorithms rely on nodes being location-aware (having location sensors that determine a node's coordinate location).

One significant difference between these two approaches is that [8] requires communication between nodes across the entire region, where as [19] relies on communication only at the region boundary. This difference has important efficiency implications, since a region containing r nodes is expected to have $\log r$ boundary nodes. (Note that this assumes the boundary measured by the geosensor network is non-fractal; this assumption is reasonable for any granular approximation of the boundary, even in cases where the underlying region is fractal, cf. [5]). Communication complexity (e.g., number of messages communicated as a function of network size) is the

overriding computational constraint in resource-limited geosensor networks. Consequently, reducing the number of nodes that must communicate, and so the total number of messages sent and received, increases the efficiency of a decentralized algorithm.

4.2 Detecting the Structure of Complex Sets of Regions

Recent work has described an algorithm for detecting the structure of complex sets of regions, and specifically the containment relationships between connected components of these objects [6]. Building on the algorithms above, [6] first computes the orientation of each boundary, using an existing decentralized algorithm for computing the area of a region [20]. Then, a single message from each boundary and marked with an identifier from the origin boundary follows a random walk through the network. At each hop, the message is updated with information about any boundaries it crosses. Using the boundary orientation, a node can locally determine whether the message is crossing into or out of a region. Tracking these changes over the message route enables a node to locally determine when it is at the boundary of a region that contains the boundary from which the message originated. When the containing boundary is crossed, the information about the contained region identifier is stored at the containing region boundary. Having stored this information, the message can then be discarded.

Figure 9 summarizes the structures used in the existing algorithms described above. Boundary nodes (black) detect the region (in gray) and have immediate one-hop communication neighbors (connected by an edge) that do not detect the region. Figure 9 presents only the simplifying case of a network structured as a maximally connected planar graph, where each boundary node neighbors two other boundary nodes, although other network structures are also allowable (cf. [8, 19]). Computing the area of each region component enables determination of consistent boundary orientation (e.g., anticlockwise, *a, b, c, ... a* and *x, y, z, ... x*). Each boundary node stores only the identity of its next neighbor anticlockwise in the cycle. (Note, dual exterior boundary structures may also be defined, but are omitted from Figure 9 for simplicity).

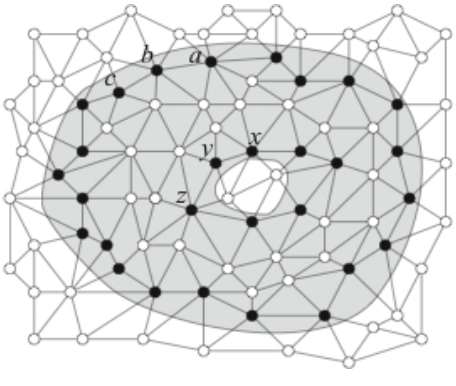


Fig. 9. Boundary structures in a geosensor network monitoring a region with one hole

4.3 Extension to Broad Boundaries

A natural way to extend the algorithms for crisp boundary detection and tracking to broad boundaries is simultaneously to track the two extremes of the broad boundary (i.e., the interface between the broad boundary and "outside" the region, and the interface between the broad boundary and "inside" the region). To avoid confusion, we refer to these extremes as "interfaces." Based on the existing algorithms, we can assume that the interfaces for all the regions' broad boundaries have been constructed, the orientation has been computed, and the nodes at the interface updated with this information as described above. Figure 10 summarizes the information constructed, alongside with the complete (centralized) containment tree. Note that the information contained in the full containment tree is decentralized, with each interface storing only the identities of its contained regions.

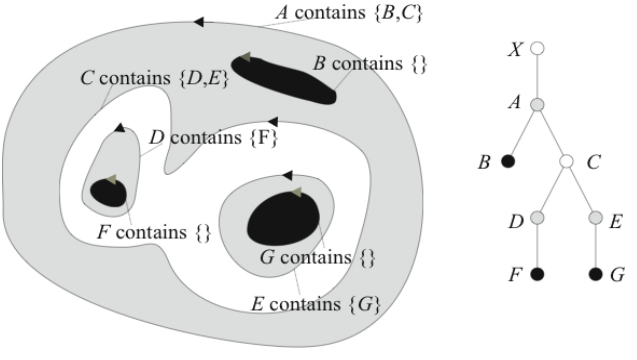


Fig. 10. Interface structure and decentralized containment tree

Three operations are then required by a decentralized algorithm for monitoring changes to the broad boundary region topology:

1. Routing a message from a contained interface A to its unique containing interface (*parent*(A)).
2. Adding or removing interface identifiers to the list of contained interfaces (*child*(A)) stored at a containing region interface A.
3. Discovering and updating the list of contained interfaces, *child*(A), of an interface A.

In general, operations 1 and 2 above are expected to have communication complexity $O(\log N)$, where N is the number of nodes in the network, since they require only routing along a boundary or other path through the network. By contrast, operation 3 requires routing through an entire region, and so is much less efficient in the worst case leading to $O(N)$ communication complexity.

Table 1 shows how these operations are applied to each of the six atomic topological changes introduced in Section 2. The table provides the name for the change (e.g., appear, disappear, ...); the interfaces that exist before and after the change (e.g., for a disappearance, interface A before leads to no interfaces $\varnothing$ after the change; for a self merging interface, interface A before still exists after the self merge, but has also

Table 1. Decentralized operations required for monitoring qualitative changes to regions with broad boundaries

Change	Before	After	Algorithm steps	Communication complexity
Appear	∅	*A*	Route message from *A* to *parent*(*A*) Remove *A* from list of *parent*(*A*) children	O(log*N*)
Disappear	*A*	∅	Route message from *A* to *parent*(*A*) Add *A* to list of *parent*(*A*) children	O(log*N*)
Merge	*A, B*	*C*	Update list of *parent*(*C*) children to be *parent*(*A*) ∪ *parent*(*A*) Route message from *C* to *parent*(*C*) Add *C*, remove *A, B* from list of *parent*(*C*)	O(log*N*)
Self-merge	*A*	*A, B**	Add *B* to list of *A* children Discover and update list of child(*B*) by flooding message through interior *B*	O(*N*)
Split	*A*	*B, C*	Route message from *B, C* to *parent*(*A*)** Remove *A*, add *B, C* to list of *parent*(*A*) children Discover and update list of *child*(*B*), *child*(*C*) by flooding message through interiors *B, C*.	O(*N*)
Self-split	*A, B**	*A*	Remove *B* from list of *A* children Route message from *A* to *parent*(*A*) Add *child*(*B*) to list of *parent*(*A*) children	O(log*N*)

* parent(B) = A ** parent(A) = parent(B) = parent(C)

enveloped a new interface *B*); and the algorithm steps required to maintain the system state (where each boundary node stores a list of the identities of its contained interfaces), based on the three operations above.

As an example, Figure 11 illustrates the operations required to update the decentralized containment tree structure following a split at interface *C*, to form two new interfaces identified *H* and *I*. Note that at the split interface C does not have information about

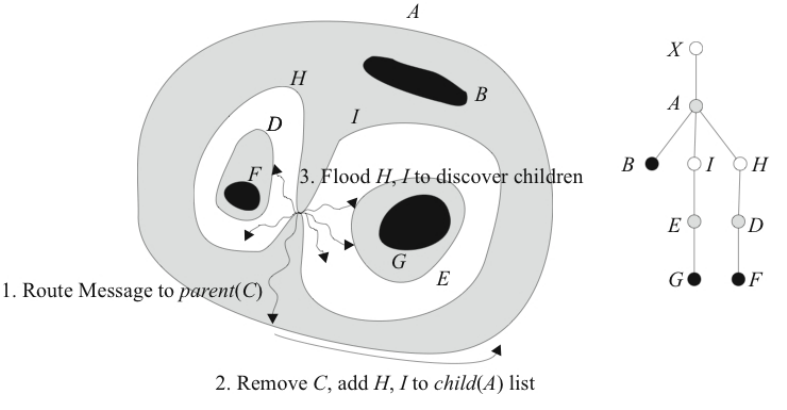


Fig. 11. Example update operations required to monitor splitting of interface *C* (cf. Figure 10)

which of the contained interfaces of *C, D* and *E*, are contained which of the two new regions *H* and *I*, necessitating the $O(N)$ discovery operation, flooding throughout *H* and *I* .

All the operations used in the algorithm have complexity $O(\log N)$, with the exception of the contained region discovery in self-merge and split, which require $O(N)$ communication complexity. Thus, the efficiency of the algorithm is in the worst case $O(N)$. However, depending on the relative proportions of self-merges and splits that occur to the broad boundary region being monitored, average case complexity may be closer to $O(\log N)$.

5 Conclusions

In this paper we have presented research that extends the theory of dynamic relationships between crisp regions to that of 3-valued regions. The domain examples provided demonstrate the need for this extension to the existing studies of the Boolean case. The approach to modeling such regions goes beyond the purely topological study of relationships between crisp regions and is founded upon the notion of the BB containment tree. We have shown how these trees are a generalization of those used previously by demonstrating that the well-known purely topological description can be obtained by a contraction process.

The second main contribution has been the demonstration of how previously developed decentralized algorithms can be composed to produce algorithms for boundary detection and tracking of regions with broad boundaries. This will be important for further developments of the theory as it provides a means to evaluate theoretical models against the behavior of wireless sensor networks which are an important source of practical examples of regions with broad boundaries.

For future work, we will show that the algorithms are scalable to large datasets by means of simulation experiments. In real life applications, 3-valued regions only provide a second approximation and, therefore, do not cover such issues as fuzziness and higher order vagueness. For more realism, we will develop an account of *n*-valued dynamic regions for *n* greater than 3. A third direction would be to model more complex spatial structure within the regions.

Acknowledgments. The work of John Stell and Michael Worboys was supported by EPSRC (EP/F036019/1) and Ordnance Survey project 'Ontological Granularity for Dynamic Geo-Networks'. Michael Worboys was also supported by the National Science Foundation under NSF grants IIS-0916219, IIS-0429644, IIS-0534429, and DGE-0504494. Matt Duckham's research is funded under an Australian Research Council Future Fellowship (project number FT0990531).

References

1. Ahmed, N., Kanhere, S., Ija, S.: The Holes Problem in Wireless Sensor Networks: A Survey. Mobile Computing and Communications Review 9(2), 4–18 (2005)
2. Antoine-Santoni, A., Santucci, J., de Gentili, E., Costa, B.: Using Wireless Sensor Network for Wildfire Detection. A Discrete Event Approach of Environmental Monitoring Tool. In: Proceedings of the First International Symposium on Environmental Identities and Mediterranean Area (ISEIMA 2006), pp. 115–120. IEEE Computer Society Press, Los Alamitos (2007)

3. Automated Local Evaluation in Real-Time—ALERT (accessed on 01/10/2010), http://www.alertsystems.org
4. Cohn, A., Gotts, N.: The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. In: Burrough, P., Frank, A. (eds.) Geographic Objects with Undetermined Boundaries, pp. 171–187. Taylor and Francis, Bristol (1996)
5. Duckham, M., Drummond, J.: Assessment of error in digital vector data using fractal geometry. International Journal of Geographical Information Science 14(1), 67–84 (2000)
6. Duckham, M., Nussbaum, D., Sack, J.-R., Santoro, N.: Efficient, decentralized computation of the topology of spatial regions. IEEE Transactions on Computers (submitted)
7. Duckham, M., Nittel, S., Worboys, M.: Monitoring dynamic spatial fields using responsive geosensor networks. In: ACM-GIS 2005, Bremen, Germany, pp. 51–60 (2005)
8. Farah, C., Zhong, C., Worboys, M., Nittel, S.: Detecting topological change using a wireless sensor network. In: Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 55–69. Springer, Heidelberg (2008)
9. Ibrahim, Z., Tawik, A.: Spatio-temporal Reasoning for Vague Regions. In: Canadian AI Conference, pp. 308–321 (2004)
10. Ibrahim, Z., Tawik, A.: A Qualitative Spatio-temporal Abstraction of a Disaster Space. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) ICWE 2007. LNCS, vol. 4607, pp. 274–281. Springer, Heidelberg (2007)
11. ISO. ISO/TC 211/WG 2, ISO/CD 19107 Geographic information—spatial schema. Technical report, International Standards Organization (2003)
12. Jiang, J., Worboys, M.: Detecting Basic Topological Changes in Sensor Networks by Local Aggregation. In: 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM GIS 2008 (2008)
13. Jiang, J., Worboys, M.,, M.: Event-based topology for dynamic planar areal objects. International Journal of Geographic Information Science 23(1), 33–60 (2009)
14. Li, Y., Wang, Z., Song, Y.: Wireless Sensor Network Design for Wildfire Monitoring. In: Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA 2006), Dalian, China, June 21-23, pp. 109–113. IEEE, Los Alamitos (2006)
15. Lynch, N.: Distributed Algorithms. Morgan Kaufmann, San Mateo (1996)
16. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless Sensor Networks for Habitat Monitoring. In: First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002), Atlanta, GA (2002)
17. Muller, P.: Topological Spatio-temporal Reasoning and Representation. Computational Intelligence 18(3), 420–450 (2002)
18. Randell, D., Cui, Z., Cohn, A.: A Spatial Logic Based on Regions and Connection. In: Nebel, B., Rich, C., Swartout, W. (eds.) KR 1992. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, San Mateo, pp. 165–176. Morgan Kaufmann, San Francisco (1992)
19. Sadeq, M.J.: In. network detection of topological change of regions with a wireless sensor network. PhD thesis, University of Melbourne (2009)
20. Sadeq, M.J., Duckham, M.: Decentralized area computation for spatial regions. In: GIS 2009: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, pp. 432–435 (2009)
21. Umer, M., Kulik, L., Tanin, E.: Spatial Interpolation in Wireless Sensor Networks: Localized Algorithms for Variogram Modeling and Kriging. Geoinformatica 14, 101–134 (2010)
22. Werner-Allen, G., Johnsnon, J., Ruiz, M., Lees, J., Welsh, M.: Monitoring Volcanic Eruptions with Wireless Sensor Network. In: Proceedings of the second European Workshop on Wireless Sensor Networks, EWSN 2005 (2005)

23. Volcano Sensorweb, http://sensorwebs.jpl.nasa.gov/ (accessed on 01/10/2010)
24. Worboys, M.F., Duckham, M.: GIS: A Computing Perspective, 2nd edn. CRC Press, Boca Raton (2004)
25. Worboys, M.F., Duckham, M.: Monitoring qualitative spatiotemporal change for geosensor networks. International Journal of Geographical Information Science 20(10), 1087–1108 (2006)

Collaborative Generalisation: Formalisation of Generalisation Knowledge to Orchestrate Different Cartographic Generalisation Processes

Guillaume Touya, Cécile Duchêne, and Anne Ruas

Laboratoire COGIT, IGN, 73 avenue de Paris, 94165 Saint-Mandé Cedex, France
{guillaume.touya,cecile.duchene,anne.ruas}@ign.fr

Abstract. Cartographic generalisation seeks to summarise geographical information from a geographic database to produce a less detailed and readable map. This paper deals with the problem of making different automatic generalisation processes collaborate to generalise a complete map. A model to orchestrate the generalisation of different areas (cities, countryside, mountains) by different adapted processes is proposed. It is based on the formalisation of cartographic knowledge and specifications into constraints and rules sets while processes are described to formalise their capabilities. The formalised knowledge relies on generalisation domain ontology. For each available generalisation process, the formalised knowledge is then translated into process parameters by an adapted translator component. The translators allow interoperable triggers and allow the choice of the proper process to apply on each part of the space. Applications with real processes illustrate the usability of the proposed model.

Keywords: cartographic generalisation, constraints, ontology, interoperability.

1 Introduction

Cartographic generalisation is a process that seeks to summarise and characterise geographical information from a geographic database in order to produce a less detailed and readable map. Automatic generalisation processes were necessary to ease the production of map series and are growingly required nowadays with the development of on-demand mapping. Many automatic generalisation methods were developed in the past years but none is actually able to tackle all the problems raised by thematic and landscape heterogeneity present in a map [1]. Rather than developing another process that would try to solve all problems of the generalisation of a map, we believe that trying a collaborative approach is a better solution. The aim of this work is to make the available generalisation processes collaborate by generalising only the part of the map they are good at. To simplify, we want to know when, where, how and why to apply a generalisation process. When developing the first generalisation processes, research already tackled these questions concerning the sequencing of atomic algorithms [2], but the problems raised are quite different at the process level.

The paper deals with one aspect of the solution of the generalisation process sequencing problem: making different generalisation processes interoperable to be

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 264–278, 2010.

sequenced in neighbouring or identical part of cartographic space may be difficult. For instance, in the agent-based process of [3], cartographic constraints, that express the map specifications, are translated in objects with methods to monitor the process, in the least squares process of [4], the constraints are translated into equations to monitor the process and the road selection process of [5] is monitored by a big set of parameters. In order to deal with this heterogeneity of inputs and make the processes interoperable, we propose to formalise specifications and cartographic knowledge as (1) constraints, (2) rules sets, (3) ontology and (4) process descriptions.

The second section of the paper presents briefly the collaborative generalisation model we propose to optimise the sequencing of generalisation processes. The third section deals with the formalisation of the cartographic knowledge to enable the collaboration between processes. Then, the fourth section explains how the formal knowledge is used to orchestrate the processes. Finally, the last section draws some conclusions and details ongoing work on the proposed collaborative model.

2 A Collaborative Approach for Generalisation

2.1 Definition of Collaborative Generalisation

We define collaborative generalisation as an approach that makes generalisation processes collaborate to generalise the part of the space they are relevant for (Fig. 1). The data to generalise is partitioned in spaces adapted to the available processes (urban areas, rural areas, mountain areas and road network in Fig. 1).Then, each space is generalised by the most appropriate process, the mapping being guided by knowledge on automated cartography, on the user specifications and on the processes capabilities. The side effects at the generalised parts neighbourhood are monitored along the whole collaborative process. Indeed, if "process 4" of Fig. 1 displaces the road network after the other three processes were triggered, it may cause new overlap conflicts with already generalised buildings and such conflicts have to be corrected.

We choose the notion of *Collaboration* in analogy with Multiagent Systems where there is collaboration when the agents share a common goal and coordinate to achieve it [6]. We consider that the Collaborative Generalisation approach makes the processes collaborate to reach the common goal of a well generalised map.

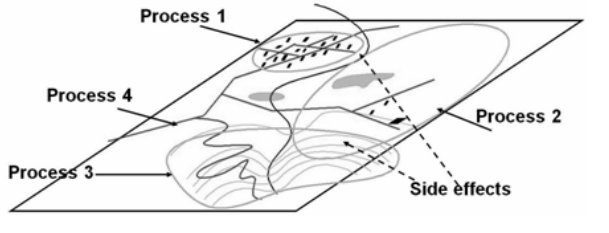


Fig. 1. The collaboration principle between generalisation processes. A process 1 is carried out on the town area, etc. Side effects are corrected at the neighbourhood of application spaces.

2.2 The Issues Related to Collaborative Generalisation

Several problems are raised by the collaborative generalisation approach. The first question concern the partitioning of the space into portions relevant for a generalisation process that we call geographic spaces: it is necessary to define the relevant spaces for each available process, their relevant boundaries, and to develop algorithms to create automatically the outline of the space. Moreover, such an approach requires to model what happens at the boundaries of the generalised geographic spaces: side effects have to be monitored. It is also necessary to find a method to reach the relevant sequence to apply.

Furthermore, manual and automated cartographic generalisation require treatment homogeneity over the map. The use of different processes to generalise a complete map could jeopardise homogeneity so the collaborative generalisation approach has to take care of this issue.

Finally, some problems of collaborative generalisation are due to the use of different processes that were not developed for working together. This issue is close to the problem of designing a generalisation process based on web services [7]. Thus, it is necessary to know how the underlying model can enable the sequencing of processes with different inputs and outputs. [8] proposes a method two combine three generalisation processes into one model and highlights the issue interoperability between generalisation processes modelled differently.

2.3 Necessary Components for a Collaborative Generalisation Approach

We propose to divide the collaborative generalisation approach in five main components and three main resources (Fig. 2): the *partitioning*, *side effects*, *scheduling*, *registry* and *translator* components and the *geographic spaces*, *formalised generalisation knowledge* and *available processes* resources. This subsection describes and illustrates these components and resources. We define a resource as the required elements that can be considered as inputs of the generalisation as they are used by the Collaborative Generalisation process or guide it. We define a component as an element that is acting in the Collaborative Generalisation and that uses resources as inputs and outputs.

The *available generalisation processes* are the generalisation processes that are accessible from the software platform where the Collaborative Generalisation model is implemented. The processes can either be implemented on the same platform as the model or called as web services, as in [7].

The *geographic spaces* are the portions of initial data that are relevant for generalisation processes and that help to process large amounts of data [1]. These spaces can be metric (i.e. a limited part of earth) as the urban or coastal areas, thematic as the road network (relevant space e.g. for the elastic beams [9]) or mixed as the mountain roads. The geographic spaces do not necessarily form a partition and often overlap as a rural and a mountain space.

The *partitioning* component is composed of spatial analysis algorithms capable of delimiting the spaces as in [10]. The partitioning component allows creating the relevant geographic spaces at the beginning of the collaborative process. The partitioning component notably requires to know which are the spaces that are useful to computed

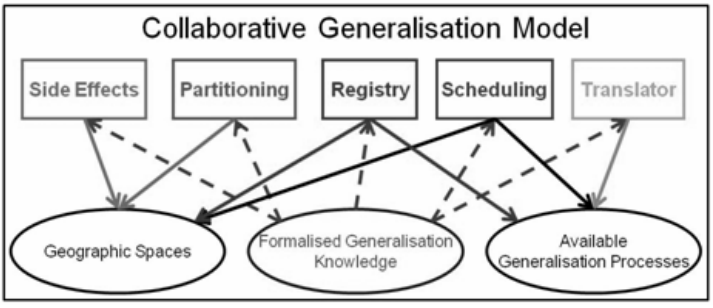


Fig. 2. The main Components (rectangles) and Resources (ellipses) of a Collaborative Generalisation Model and how the components act on the resources (plain arrows). The Formalised Generalisation Knowledge is used by all five components. The Side Effects and the Partitioning components act on Spaces while the Registry and the Iterating component act on both Spaces and Processes and the Translator only acts on Processes.

according to the user specifications and the available processes. Such knowledge is included in the *formalised generalisation knowledge* resource.

The geographic spaces being identified, we define a sequence of collaborative generalisation as a list of pairs (geographic space, generalisation process) interrupted by side effect processes. For instance a collaborative sequence could be: (Urban space 1, Process 1), (Urban space 2, Process 1), (Rural space 1, Process 2), side effects correction in Rural space 1 neighbourhood, (Mountain space 1, Process 3)...

The *registry* component aims at matching the pairs as yellow pages answering the question: what is the process to generalise this space? The registry records the services that the *available generalisation processes* are able to provide. Then, when a geographic space requests for generalisation, the registry component answers with a list of relevant processes. The registry mechanism is detailed more in section 4.4. The registry component clearly requires a description of the generalisation processes capabilities and needs to have access to user specifications to decide the application relevance of a process, both included in the *formalised generalisation knowledge* resource. The formal description of the generalisation processes capabilities is detailed in section 3.6.

The *scheduling* component chains the pairings of spaces and processes in an optimal sequence. It decides at each step which space has to query the registry for generalisation and evaluates the generalisation results. To iteratively choose the next space to be generalised, the scheduling component requires both user specification and general knowledge on the major steps of generalisation. The sequence is not linear but optimised by a trial and error strategy guided by general knowledge and online evaluation.

The *side effects* component relies on the observation of the neighbourhood of the spaces generalised by the *scheduling* component. The component monitors the potential side effects by triggering a deformation process as [11] that reduces conflicts without undoing the previous generalisations. The component requires to know user specifications in order to maintain the ones that are altered by side effects.

The *translator* component parameterises the available processes according to the user specifications whatever the process parameterisation system is. The translator component is detailed in sections 4.1 to 4.3.

Finally, the *formalised generalisation knowledge* resource gathers user specifications, generalisation processes descriptions and general knowledge on the scheduling steps of generalisation. We developed a Collaborative Generalisation model that relies on the components and resources described in this section, the *CollaGen* model (for Collaborative Generalisation). This paper focuses on the *formalised generalisation knowledge* resource modelling in *CollaGen*, presented in the next section. The interactions between the formalised knowledge and the translator and registry components are described in the fourth section.

3 Formalisation of Generalisation Knowledge

3.1 Organisation of the Formalised Generalisation Knowledge

In order to provide knowledge to the CollaGen model, user specifications and knowledge on cartographic generalisation are formalised in a machine interpretable way. User specifications cover here both the user requirements for the generalised map and the cartographic rules for map legibility. The formalised knowledge required for collaborative generalisation can be divided in five parts: generalisation domain ontology, generalisation constraints set, operation rules set, sequencing rules set and process descriptions.

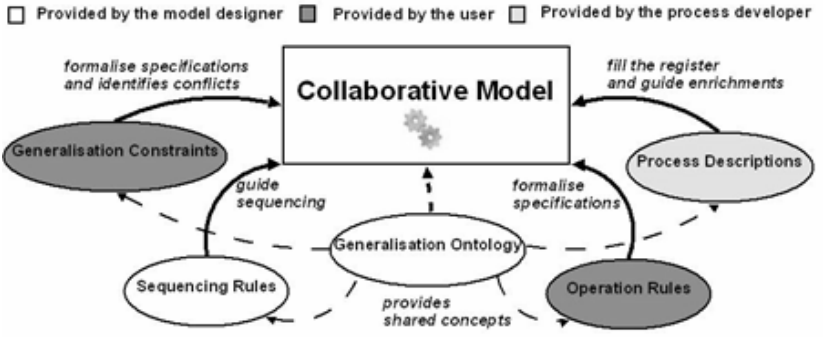


Fig. 3. A diagram of the 5 parts of formalised knowledge and their use in the collaborative model. The dashed arrows show that the Ontology provides shared concepts to every part.

Fig. 3 shows how this formalised knowledge is organised to feed the collaborative process. The formalised knowledge is generated by three actors of the collaborative model that correspond to three times in the model life: the model designer that implements the five components and designs the generalisation ontology and the sequencing rules; the process developer that makes generalisation processes available and describe them, enriching potentially the ontology; the user that aims at generalising his data and then translates his specifications into generalisation constraints and operation rules.

The five following sections describe in detail the formalisation model of each piece of formalised knowledge and explain how the models are instantiated.

3.2 A Generalisation Domain Ontology

Automatic cartographic generalisation requires as input data an adapted data schema [12]. The adapted data schema is the initial schema of the geographic database used to produce the generalised map, enriched with implicit concepts made explicit in the data to allow the automatic process. The implicit concepts useful for automatic generalisation can be of different kinds: *meso* concepts [3] like "group of building", "city" or "highway interchange"; *procedural* concepts that are necessary for the use of a particular process like the "fields" for a GAEL process [11], "dead ends" for a road selection process or "small compacts" for a CartACom process [3]; explicit *geographic relations* like the proximities between objects or the accessibility of a facility by a road. The generalisation constraints that mostly formalise the user specifications may concern the concepts and data added in the adapted schema.

We define a generalisation domain ontology as a domain ontology concerning the automatic generalisation process. The generalisation domain ontology should be made of:

- The concepts that can be present in an adapted data schema (meso, procedural concepts and geographic relations plus topographic concepts).
- The known relevant geographic spaces.
- The properties that may be constrained by user specifications.
- Generalisation operator taxonomy.
- A taxonomy of the generalisation processes available on the platform.

The geographic properties of concepts that are likely to be constrained by user specifications are included in the ontology as ontology properties. For instance, "area", "granularity" or "absolute_position" are some of the properties defined on the "building" concept while "sinuosity", "length" and "coalescence" are some "road" properties because constraints are often defined on these properties. Properties are also defined on the geographic relations: the "proximity" relation as a "minimum distance" property. The modelling of properties as results of spatial analysis measurements is advanced. Defining that shape should be measured by a mix of "compactness", "concavity" and "elongation" properties is not possible yet. Describing more in detail the properties as in [13] would help to make a direct link between the atomic properties and the spatial analysis methods to measure the atomic properties and more abstract ones. Associations related to the adapted schema are also included in the ontology (e.g. the association "a *meso_entity* is composed of *geographic_entities*"). Some restrictions are defined on the associations. For example, the meso composition association can be restricted for building groups to only buildings and roads.

Our implementation of the generalisation domain ontology, in OWL 2, is built upon a topographic database concept taxonomy that was originally created in OWL by an automatic natural language process [14], manually enriched by the properties, associations and new concepts necessary to produce the generalisation domain ontology. The concepts possibly present in the adapted schema where classified using the national mapping experience of the laboratory. The generalisation operator taxonomy chosen as the most relevant for this ontology is extracted from [15] Then, the

well-known meso, procedural and relations were added to the ontology with specific associations like a *meso_entity* "is composed of" *geographic_entities*.

The generalisation domain ontology is used as the support of generalisation knowledge sharing and integration, which is one of the applications of ontology [16].

3.3 Formalisation of Generalisation Constraints

In the first years of cartographic generalisation research, constraints have quickly been considered as the best way to formalise the map specifications [17]. Indeed, constraints, like "inter-distance between buildings must be at least 0.1 map mm", are a convenient way to express the legibility conditions of a map. Generalisation constraints classifications were also suggested [3, 18]. Several research or production projects have proposed models to capture the user specifications in the form of generalisation constraints using table templates or OCL expressions [18, 19] while commercial software like Clarity™ (1Spatial) or Axpand® (Axes Systems) propose ad-hoc constraints expression models.

We developed a model to express the different user and map specifications as constraints that rely on the referred models and classifications (Fig. 4). Four types of constraints are defined from the classification of [18]: *micro* constraints (constraints on single objects), *meso* constraints (constraints on group of objects or patterns) and relational constraints (constraints on the geographic relation between two objects) and the macro constraints (constraints on the population of all objects of a kind). Only the last type is not present in the classification of [3]. Our contribution is the rest of the formal model described in Fig. 4. The model is described using the two following constraints examples:

- C1: "Buildings' area must be over 0.2 map mm² in urban areas".
- C2: "Very concave buildings should maintain initial concavity with 10% margin"

The major properties of a constraint are its *name*, and the *concept* ("building" for C1 and C2) and *character* constrained ("area" for C1 and "concavity" for C2) from the ontology. Then, generalisation constraints are characterised by an *expression type*, a *selection criterion* and a *space restriction*. The *expression type* is an object that holds

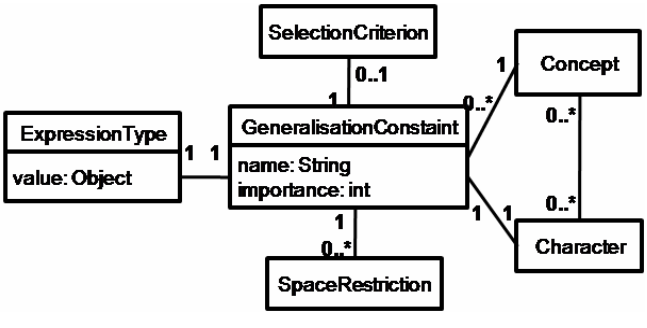


Fig. 4. UML class diagram of the Generalisation Constraints formal model. A constraint concerns a concept and one of its characters from the ontology and has an expression type, a selection criterion and a space restriction.

both the kind of expression of the constraint and the threshold values. For example, the “threshold” type of expression means that the constraint is like: “concept.character < value”. Thus C1 has a "threshold" expression type with ">" as operator, "0.2" as value and "map mm²" as unit. C2 has a "margin" expression type with a "10%" value. Five kind of *expression types* have been defined. The *selection criterion* is a query that selects one part of the objects of the constrained concept. For instance, C2 has a *selection criterion* that queries only very concave buildings (a threshold has to be given to translate "very" in understandable concavity value). The selection criterion can be seen has an implementation of the OGC Filter standard [20] for constraints. The *space restriction* is a set of geographic spaces from the ontology where the constraint is only applied. When the set is empty, the constraint concerns every part of space. Only C1 has a space restriction as the constraint is only valid in urban spaces.

A Graphical User Interface (GUI) form has been developed to help the user capture the constraints and implement the formal model. 70 constraints have been captured, extracted from French NMA experience.

3.4 Formalisation of Operation Rules

Although generalisation constraints may express most of user specifications, some part of the specifications cannot be appropriately expressed by generalisation constraints. Indeed, some of the constraints extracted from the EuroSDR test [18], particularly the ones advising or forbidding actions to apply ("...buildings should be aggregated"), are clearly rules that were forced to fit in the constraints template. So, we consider that it is simpler for a user to express them as Operation Rules. The rules are modelled following equation (1):

$$Premise_1 \wedge Premise_2 \wedge ... \Rightarrow Conclusion \quad (1)$$

Conclusions are generalisation operations from the ontology that are advised or not (e.g. “*Roundabout diameter < 100 m* implies *Collapse to point*”). A premise is a simple condition expressed with a threshold on a concept property, as in "threshold typed" constraints introduced in 3.3. Operation Rules can be seen as a convenient vector for modelling systematic operations as in the above roundabout rule. Operation rules are also a way to guide generalisation processes in their actions: the rule “buildings should not be aggregated in urban spaces” helps to parameterise the generalisation process that will be chosen to generalise urban spaces.

3.5 Formalisation of Sequencing Rules

The CollaGen model allows the expression of "sequencing rules" that represent general knowledge in automated cartographic generalisation and provide general guidelines to the scheduling component. The sequencing rules correspond to the *Global Master Plan* described in [21]. The Global Master Plan described how the main steps of generalisation are chained. For instance, the well-known rule “Network selection must be carried out before cartographic generalisation” can be expressed and processed thanks to sequencing rules.

As operation rules, the sequencing rules are modelled using premises and a conclusion. Fig. 5 shows the model of sequencing premises and conclusions. Premises refer to a particular place in the sequence of generalisation processes: “after network

selection" or "when each part of the space has been processed once at least" are instances of particular places in the sequence. Conclusions can be either a geographic space ("Urban spaces should be processed first in cartographic generalisation") or a process ("Geometry Collapses should be processed first") from the ontology.

The implemented sequencing rules allow to sequence the generalisation process in four main steps: the geometry type changes (e.g. collapse of roundabouts to points), the selection (elimination of useless objects), the cartographic generalisation and the graphic generalisation [4] (correction of remaining legibility conflicts).

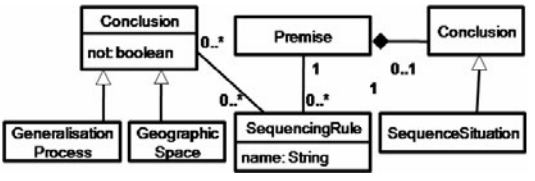


Fig. 5. The UML data schema of the Sequencing Rules model. A premise is a situation in the processes sequence and the conclusion is a generalisation process or a geographic space.

3.6 Formal Description of Generalisation Processes

As an analogy to web service composition, the composition of generalisation processes requires the description of their capabilities and requirements. The relevance domain of the different generalisation processes has to be formalised to know where they can be applied. For instance, we should be able to say that the CartACom process [3] is relevant on rural spaces or low density spaces and that the Elastic Beams [9] are relevant on flexibility graphs [22] (conflicting sub-graphs of the road network adapted to the Beams). We should also formalise which constraints can be handled by a process in order to know if it is adapted to particular situation. Regarding web service composition, the description of the service capabilities can be formalised by pre-conditions and post-conditions [23]. The pre-conditions correspond to the conditions the input data have to meet to be properly processed. The post-conditions describe the expected data modifications caused by the process. In the CollaGen model, this model is followed to describe the capabilities of generalisation processes in our collaborative model where pre-conditions are the relevant spaces for application and the post-conditions are the a priori handled constraints and rules (Fig. 6). Pre-conditions refer to spaces described in the ontology and post-conditions refer to constraints and rules present in the sets of constraints and rules defined by the user.

To go further in the process description details, some properties are associated to the processes among which the generalisation method the process is an instance of (e.g. "AGENT specialised for urban generalisation" is an instance of "AGENT model"). It enables to link this process to the Sequencing Rules. The name of the programming component that allows to execute the process, is mentioned ("name-Java" attribute of class ProcessDescription in Fig. 6), which is a way of distinguishing *function* and *component* [24]. Added to that, the *scale range* class allows to define for the process the initial and final scales for an appropriate use of the process (e.g. the urban specialised AGENT process is appropriate for 1:10k to 1:50k). The limit scale ranges are also included in the class. Moreover, the required data enrichments to run

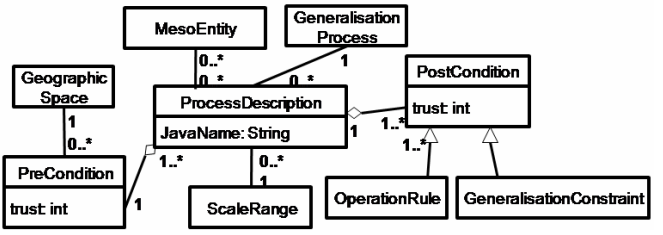


Fig. 6. UML class diagram of the generalisation process description for interoperability between processes. The pre-conditions are the spaces where the process is applicable and the post-conditions are the rules and constraints a priori satisfied after process execution.

the process (e.g. “dead-ends”, “road partition” or “building alignments”) are described in terms of meso or procedural ontology concepts that are expected to be added in the data. If the process is chosen by the *registry* component to generalise a given geographic space, the first step is then to process the enrichments on the space. Finally, the *trust* attribute on both *PreCondition* and *PostCondition* classes (Fig. 6) is an a priori evaluation of the relevance of each condition, provided by the process provider.

A GUI helps the process provider to fill the description that is automatically translated into the CollaGen description model. Eight generalisation processes available on our research platform are described including AGENT [3], CartACom [3], least squares [4], GAEL [11], elastic beams [9] and a road geometry collapse [5].

4 Processing Generalisations from Formal Knowledge

This section describes how the formal knowledge is used in the model by the *translator* and *registry* component. Section 4.1 deals with the need for matching the data schema to the ontology. Section 4.2 shows how the use of translator functions allows to trigger interoperable generalisations. Section 4.3 explains how, for a given geographic space, the relevant generalisation process is chosen. Some automatically triggered generalisation results illustrate the CollaGen model in section 4.4.

4.1 Matching Data Schema to the Ontology

Linking information resources (a geo-database schema here) to an ontology is made through a process called *annotation* [25]. In the CollaGen translator component, we used the annotation method called *registration mapping* that is a separate source containing the matching between schema elements and ontology concepts [25]. In the registration mapping, the useful ontology concepts, properties and associations are mapped to the equivalent in the data schema. For instance, the concept "road" is mapped to the class "BD_TOPO_Road_Section". We define the useful concepts as the ones that are actually used in the collaborative process (referred to in the constraints, operation rules and process descriptions).

Making the registration mapping automatically would require natural language processes that are not priority of this research so we opted for an interactive method. For instance, a test case with one process, "urban AGENT" that requires the enrichment

with building groups and three constraints on building minimal area, minimal granularity and inter distance, requires several mappings: first the ontology concepts "building" and "building group" have to be mapped to classes of the data schema; then the properties "area", granularity" on "building" and "building inter distance" on "building group" have to be mapped to the attributes of the data schema.

4.2 Translating Knowledge into Process Parameters

Once the objects of the database are matched to the ontology thanks to the registration mapping, the link between the objects and the constraints related can be made and so generalisation can be triggered. The generalisation processes first need to be parameterised according to the expressions and values held by the constraints and the rules captured by the user. As generalisation processes are very complex, they often require a big set of parameters and proper initialisations (e.g. defining constraints for AGENT, equations for the Least squares), giving importance to this translation step. We consider a process parameterised when all required parameters and initialisations have been set up. Thus, registering a generalisation process to the CollaGen model also requires providing a translator component that is able to read the constraints and translate them into the process parameters. A translator function of the component can be considered as a simple programming interface that enables the publishing of the process as a service, which is a key point of geo-processing interoperability [24].

Each generalisation process is provided with its standardised translator function (Equation 2). The body of the translator function consists in searching, for each parameter, for a constraint or rule in the sets that correspond to the parameter and in getting the value held in the constraint as the parameter.

$$\text{parameterised process} = f(p, C, R, rm). \quad (2)$$

Where p is the process to be parameterised, C is the constraints set, R is the operation rules set and rm is the registration mapping.

We developed the translator functions for the 8 generalisation processes available on our platform. For instance, the road geometry collapse process has simply real threshold parameters while the CartACom process is parameterised with constraints and the least squares process with equation systems. Fig. 7 shows two generalisation results from these three processes obtained with automatic trigger and parameterisation from the formal knowledge we captured to test our model, and the translator. A third result obtained with CartACom process is presented in Fig. 8.

4.3 Choosing the Best Process to Generalise a Geographic Space

As mentioned in section 2, the generalisation process descriptions are stored in a yellow pages registry that can be consulted to find the best process to generalise a geographic space designated by the *scheduling* component. The CollaGen implementation of the registry responds to a request with a list of relevant processes in relevance order. As in web search engines, the registry response is divided in two steps, the filter step that selects only the relevant services and the ordering step that orders the filter response in terms of relevance. In CollaGen, the filter step questions the preconditions of the descriptions (i.e. the geographic spaces a priori accepted as possible

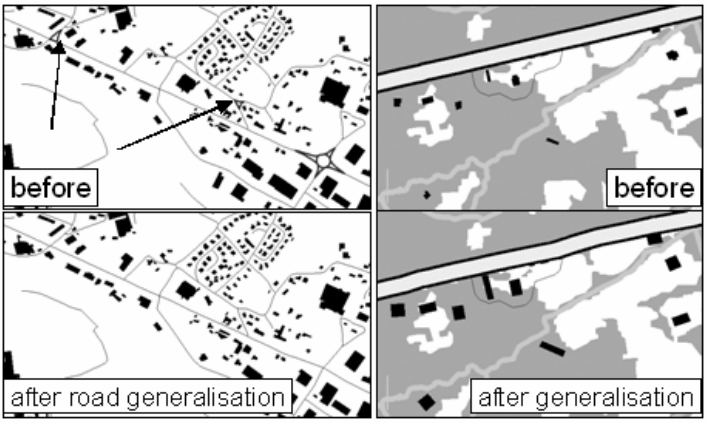


Fig. 7. Generalisation results from two different processes parameterised automatically by the formal knowledge and the translators. On the left, a road geometry collapse process parameterised with the translation of two rules concerning roundabouts and branching crossroads (highlighted with arrows). On the right, a least squares process parameterised with constraints on proximity **b**etween roads and buildings and on building size and granularity.

input for the process), and keeps the processes whose pre-conditions correspond to the space concerned by the request. For instance, if only the AGENT process and the least squares process have "urban space" in their pre-conditions, an urban space requesting a generalisation will only get these two answers. As a first approximation, the ordering step of the request is made in two times. A first ordering is made according to the "trust" value (integer between 1 and 5) of the pre-condition. Then, pre-conditions with the same trust value are ordered according to the post-conditions, that are the constraints and rules a priori satisfied. The more the post-conditions match the actual constraints conflicts, the best the process is rated.

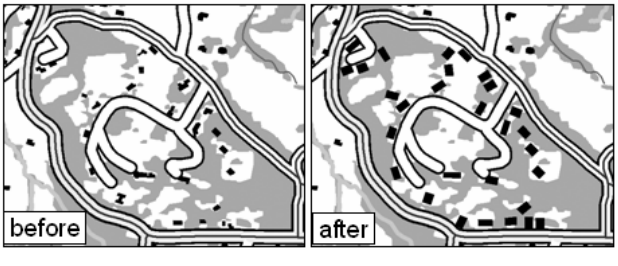


Fig. 8. A Rural space (built by the partitioning component) generalised by the CartACom process according to the request to the registry of generalisation processes.

Fig. 8 illustrates the choice of the best process to generalise a "Rural" geographic space. Three of the eight available processes have a pre-condition about rural spaces: "CartACom" (trust value of 4), "Urban AGENT" (trust value of 2) and "least squares" (trust value of 2). "CartACom" is put on top of the list and tried first. The ordering of

the two remaining ones is done comparing the conflicts in the rural space to the post-conditions. For instance, the preservation constraint "preserve parallelism between roads and buildings" causes conflicts that should be dealt by the post-conditions. Finally, the “least squares” is advised first for a better preservation of the parallelism constraint. Anyway, as the generalisation with the CartACom process is evaluated as satisfying, the following propositions in the list are not considered.

4.4 Some Results with Several Processes

Although the CollaGen model is not fully implemented some results can be presented. Fig. 12 shows several processes parameterised by the translator that are executed on the same situation. The four processes used in this example are fully interoperable within CollaGen and we can see that the third sequence gives the best results. It is hopefully the first one proposed by the registry regarding the rural space the situation is in: the registry proposes CartACom with the rural space in firstly generalised then proposes the Least Squares as the best process for final graphic generalisation [4].

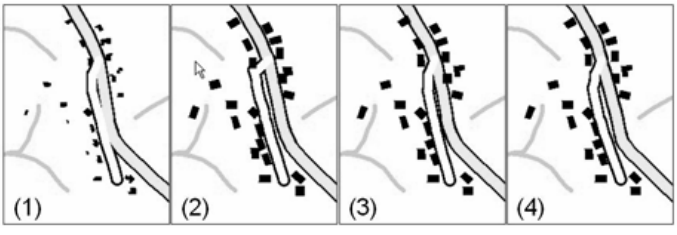


Fig. 9. (1) a situation before generalisation. (2) the situation generalised with an AGENT based process then a Least Squares process: some conflicts remain. (3) the situation generalised with a CartACom process then a Beams process: some side effects are created by the beams. (4) the situation generalised with CartACom then Least Squares: it is correctly generalised.

5 Conclusions and Further Work

The paper introduced and defined the collaborative generalisation approach. In such an approach the initial data is partitioned in different geographic spaces (cities, countryside, mountains, etc.) that are generalised by the more appropriate of the available automatic processes while side effects between spaces are controlled. We presented an important aspect of the CollaGen model (our implementation of collaborative generalisation): to enable the interoperability of the processes and the homogeneity of the generalisation, cartographic knowledge and user specifications are formalised in constraints and operation rules sets, sequencing rules and process descriptions, all based on generalisation domain ontology. Once the initial data is annotated with the ontology, translator components allow parameterising the processes and the processes can be chosen and triggered on a given geographic space.

To go further, some classical generalisation constraints could be integrated in the ontology to ease the capture of specifications by the user. But before, two topics have to be tackled more deeply to make the CollaGen model operational. First, the

management of the side effects has to be clarified: when do we exactly need to trigger the correction and how do we observe the related conflicts? Then, the scheduling component implementation has to be finalised.

References

1. Touya, G.: First thoughts for the orchestration of generalisation methods on heterogeneous landscapes. In: 11th ICA Workshop on generalisation and multiple representation, Montpellier, France (2008), http://aci.ign.fr/montpellier2008/papers/01_Touya.pdf
2. McMaster, R.B., Shea, K.S.: Cartographic generalization in digital environment: A framework for implementation in a gis. In: GIS/LIS 1988, pp. 240–249 (1988)
3. Ruas, A., Duchêne, C.: A Prototype Generalisation System Based on the Multi-Agent System Paradigm. In: Mackaness, W., Ruas, A., Sarjakoski, T. (eds.) The Generalisation of Geographic Information: Models and Applications, pp. 269–284. Elsevier, Amsterdam (2007)
4. Harrie, L., Sarjakoski, T.: Simultaneous Graphic Generalization of Vector Data Sets. Geoinformatica 6(3), 233–261 (2002)
5. Touya, G.: A Road network Selection Process based on Data Enrichment and Structure Detection. In: Transactions in GIS (in press, 2010)
6. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (2000)
7. Regnauld, N.: Evolving from automating existing map production systems to producing maps on demand automatically. In: 10th ICA Workshop on Generalisation and Multiple Representation (2007), http://aci.ign.fr/BDpubli/moscow2007/Regnauld-ICAWorkshop.pdf
8. Duchêne, C., Gaffuri, J.: Combining Three Multi-agent Based Generalisation Models: AGENT, CartACom and GAEL. In: Ruas, A., Gold, C. (eds.) Headway in Spatial Data Handling, 13th International Symposium on Spatial Data Handling. LNG&C, pp. 277–296. Springer, Heidelberg (2008)
9. Bader, M., Barrault, M.: Cartographic Displacement in Generalization: Introducing Elastic Beams. In: 4th ICA workshop on progress in automated map generalisation, Beijing (2001), http://aci.ign.fr/BDpubli/pbeijing2001/papers/bader_barraultv1.pdf
10. Chaudhry, O.Z., Mackaness, W.A.: Creating mountains out of mole hills: Automatic identification of hills and ranges using morphometric analysis. Transactions in GIS 12(5), 567–589 (2008)
11. Gaffuri, J.: Three reuse example of a generic deformation model in map generalisation. In: 24th International Cartographic Conference, Santiago, Chile (2009)
12. Ruas, A.: Automating the generalisation of geographical data: the age of maturity? In: 20th International Cartographic Conference, Beijing, China, pp. 1943–1953 (2001)
13. Lüscher, P., Weibel, R., Mackaness, W.A.: Where is the terraced house? On the use of ontologies for recognition of urban concepts in cartographic databases. In: Ruas, A., Gold, C. (eds.) Headway in Spatial Data Handling, pp. 449–466 (2008)
14. Abadie, N., Gesbert, N., Mustière, S.: Création d'une ontologie à partir des spécifications textuelles pour l'intégration des bases de données géographiques. In : 17èmes journées Ingénierie des Connaissances, Nantes (2006)

15. Mustière, S.: Apprentissage supervisé pour la généralisation cartographique. PhD thesis, Université Pierre et Marie Curie (2001)
16. Horrocks, I.: Ontologies and the semantic web. Commun. ACM 51(12), 58–67 (2008)
17. Beard, K.: Constraints on rule formation. In: Buttenfield, B., McMaster, R. (eds.) Map Generalization, pp. 121–135. Longman (1991)
18. Burghardt, D., Schmid, S., Stöter, J.: Investigations on cartographic constraint formalisation. In: 10th ICA Workshop on Generalisation and Multiple Representation, Moscow (2007),
http://aci.ign.fr/BDpubli/moscow2007/Burghardt-ICAWorkshop.pdf
19. Stoter, J.E., Morales, J.M., Lemmens, R.L.G., Meijers, B.M., van Oosterom, P.J.M., Quak, C.W., Uitermark, H.T., van den Brink, L.: A Data Model for Multi-scale Topographical Data. In: Ruas, A., Gold, C. (eds.) Headway in Spatial Data Handling, pp. 233–254 (2008)
20. Filter Encoding Implementation Specification,
http://www.opengeospatial.org/standards
21. Ruas, A., Plazanet, C.: Strategies for Automated Generalization. In: 7th International Symposium on Spatial Data Handling, Delft, Netherlands, pp. 319–336 (1996)
22. Lemarié, C.: Generalisation process for top100: research in generalisation brought to fruition. In: 5th ICA Workshop on progress in automated map generalisation (2003)
23. Lutz, M.: Ontology-Based Descriptions for Semantic Discovery and Composition of Geoprocessing Services. GeoInformatica 11(1), 1–36 (2007)
24. Bucher, B., Jolivet, L.: Acquiring service oriented descriptions of GI processing software from experts. In: 11th AGILE International Conference, Girona, Spain (2008)
25. Lemmens, R.: Lost and found, the importance of modelling map content semantically. In: Peterson, M. (ed.) International Perspectives on Maps and the Internet, pp. 377–396 (2008)

Automatic Extraction of Destinations, Origins and Route Parts from Human Generated Route Directions

Xiao Zhang[1], Prasenjit Mitra[1,2], Alexander Klippel[3], and Alan MacEachren[3]

[1] Department of Computer Science and Engineering
[2] College of Information Science and Technology
[3] Department of Geography
The Pennsylvania State University
xiazhang@cse.psu.edu, pmitra@ist.psu.edu,
{klippel,maceachren}@psu.edu

Abstract. Researchers from the cognitive and spatial sciences are studying text descriptions of movement patterns in order to examine how humans communicate and understand spatial information. In particular, route directions offer a rich source of information on how cognitive systems conceptualize movement patterns by segmenting them into meaningful parts. Route directions are composed using a plethora of cognitive spatial organization principles: changing levels of granularity, hierarchical organization, incorporation of cognitively and perceptually salient elements, and so forth. Identifying such information in text documents automatically is crucial for enabling machine-understanding of human spatial language. The benefits are: a) creating opportunities for large-scale studies of human linguistic behavior; b) extracting and georeferencing salient entities (landmarks) that are used by human route direction providers; c) developing methods to translate route directions to sketches and maps; and d) enabling queries on large corpora of crawled/analyzed movement data. In this paper, we introduce our approach and implementations that bring us closer to the goal of automatically processing linguistic route directions. We report on research directed at one part of the larger problem, that is, extracting the three most critical parts of route directions and movement patterns in general: origin, destination, and route parts. We use machine-learning based algorithms to extract these parts of routes, including, for example, destination names and types. We prove the effectiveness of our approach in several experiments using hand-tagged corpora.

Keywords: driving directions, route component classification, destination name identification, geographic information extraction.

1 Introduction

GIScience is intimately linked with a growing interest in revealing how the human mind understands spatial information [1]. In the end, analyses of spatial information have to be meaningful to humans and as such it is only natural to look into how humans make sense of information about the environment in the first place. One of the most influential revelations that brought together researchers from the cognitive science community and spatial information science is the role that metaphors play in understanding (spatial) information. The work on metaphors by Johnson and Lakoff [2] has lead to the

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 279–294, 2010.

identification of recurring patterns in human (spatial) lives, their direct and embodied experiences that shape their thoughts and that allow humans to understand abstract concepts that are otherwise hard to grasp. These recurring patterns are referred to as *image schema* [3]. Many researchers across different disciplines have built on this work recognizing the importance of identifying and using information that can be organized from an image-schematic perspective.

One of the most fundamental image schemas is that of a *path* with three distinct elements: origin, the path itself, and the destination (source - path - goal in the original terminology). The importance of understanding this seemingly simple schema is corroborated by the fact that a plethora of articles has been published on this topic across disciplinary boundaries [4] [5] [6]. From a GIScience perspective, last year's best paper award for this conference went to Kurata for his paper on the 9+ - intersection model [7]. This model explicitly adds origins and destinations to the 9-intersection model [8] allowing us to characterize the human understanding of movement patterns more adequately. In follow up articles, this topological characterization of a trajectory in relation to a spatially extended entity (a region) has been applied to interpret movement patterns of individual agents [9]. Hence, understanding where someone or something is coming from (Origin), where someone or something is going to (Destination), and how someone or something got from the Origin to its Destination are essential aspects in characterizing and interpreting movement patterns.

The importance of analyzing movement patterns (as one form of spatio-temporal information) is also reflected in current multinational research programs that bring together researchers that develop tools, methods, and theoretical frameworks for properly modeling and analyzing movement patterns [10]. While the majority of research focuses on coordinate-based information, we are interested in identifying how humans characterize movement patterns linguistically and how to interpret these descriptions automatically. Being able to develop computational systems that interpret spatial language is important for the following reasons (among others):

- Text documents are large in number and developing tools to automatically analyze them is essential to cope with what Miller called the data avalanche [11]. We are not only facing a quantitative data avalanche, but also a qualitative one.
- Automatic understanding of large amounts of text on the World-Wide-Web opens opportunities to collect data and perform (spatial) analysis.
- Spoken dialogue can be transformed into written text. Linguistic communication is the primary way for humans to exchange thoughts, ideas, and information.

More specifically, with the spatial information encoded in origins, destinations, and route parts databases can be built that allow for answering questions such as: How many linguistically coded routes ended in the Washington DC area or what linguistic patterns (e.g. relative or cardinal directions) are people in different regions of the US using.

However, the challenge of automatically analyzing linguistically encoded movement patterns, and specifically route directions, are manifold. Any automatic extraction method must deal with:

- the specifics of documents on the syntactic level (e.g., html code)
- underspecified information and/or missing information
- different linguistic styles, varying semantics, synonyms and their disambiguation.

In this paper, we propose algorithms to automatically identify destinations, origins and route parts from text documents. First we use machine-learning-based algorithms to classify the sentences extracted from route direction documents into route components (destinations, origins and route parts). Then, based on the classification results, our algorithm extracts candidates for destination names. After that, the algorithm re-examines the classification results in order to improve the identification of destinations.

The remainder of the paper is structured as follows. In Section 2, we define important concepts used in this paper and formulate our problem. We review related work in Section 3. In Section 4, we describe our proposed algorithms in detail. Section 5 contains the results of our experiments; they show the effectiveness of our algorithms. In Section 6, we conclude the paper and propose future work.

2 Preliminaries and Problem Formulation

Route directions instruct people how to travel from an origin or an area to a destination. We study direction documents that contain the following route components [12] [13] [14]: **destination**, which is the place a person travels to, often an address or the name of the place; **origin**, which is the place or area a person comes from, often a city, a (cardinal) direction or a highway name; and **instructions** (or route parts; these terms will be used interchangeably in the rest of the paper), which are a set of path segments or route segments a person should follow in order to reach the destination from the origin. Figure 1 gives an example of a driving direction Web page [1]. In this document, the destination is "*Directions to IST*"; one of the origins is "*From University Park Airport:*"; the first instruction for this origin is "*Head southwest on Fox Hill Rd. toward High Tech Rd. (2.4 mi)*". In addition to the route components, direction documents also contain information irrelevant to route directions, such as advertisements and phone numbers. They are called "**other**" in this paper.

Route components and other information are expressed in the form of a complete sentence or a stand-alone phrase. They are referred to as "sentence"s throughout this paper. Given the list of all sentences extracted from a document containing driving directions, the **first task** of our study is to classify the sentences into one of the four categories: (1) destination, (2) origin, (3) instruction or (4) other.

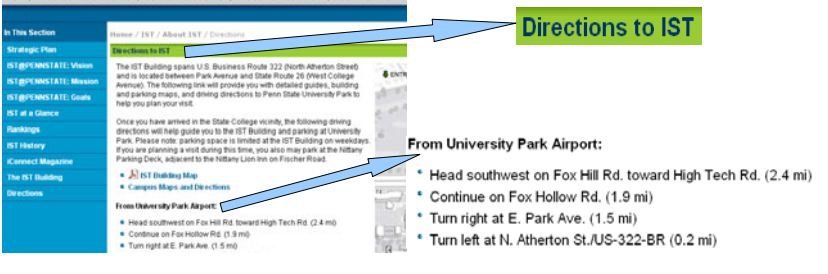


Fig. 1. An example of a direction document

[1] http://ist.psu.edu/ist/iststory/page2.cfm?pageID=1043

A destination is a very important route component because the route ends at the destination and that is where the person following the directions wants to reach. One destination can serve several routes. If the name of the business or organization providing driving directions (referred to as destination names, such as "*Evergreen Golf Club*", "*The Happy Berry, Inc.*") is successfully identified, we can find its coordinates and further locate it on a map. The type of the destination (for example "*university*", "*hospital*") is also very helpful, because if the destination can be narrowed down to a small area and all business names in this area are available, the type can help us pinpoint the destination. Therefore, the **second task** of our study is as follows: given the list of sentences in a direction document, extract the name or the type of the destination.

However, as will be shown in Section 4.2, the recognition of destinations is a difficult problem, especially for stand-alone destination names. Such sentences are often very short. They lack the features that make them stand out from the other route components, and are frequently mis-classified as "other". Without using additional information about the destination names in their context, it will be very difficult to recognize them. Based on our observation, such information can be found in the "**arrival information**" of the directions. The "arrival information" is the last sentence in a set of instructions. The name and type of the destination are often mentioned in it, for example, "*Jordan Hall is located immediately to your right.*". We proposed an algorithm to extract destination names and types from arrival information and use it to improve the recognition of destinations. With this new information, we try to accomplish our **third task:** improve the number of recognized destination sentences over all the destination sentences in the documents (also referred to as *recall* in information retrieval [15]).

3 Related Work

Our system uses machine-learning models to classify sentences extracted from the direction documents into route components. Based on our observation, the sentences displayed a strong sequential nature, for example, instructions are often grouped together; origins are often followed by instructions and other information often appears together. Therefore, we first review related work on machine-learning algorithms designed to label sequential data. Then, we review previous work on sentence classification in other domains. Our task of identifying destination names is related to but essentially different from named entity recognition problems, hence, we review this field too.

3.1 Labeling Sequential Data

Labeling sequential data involves assigning class labels to sequences of observations. Labeling sequential data includes Part of Speech (POS) tagging and entity extraction. Sequential data has: 1) statistical dependencies between the objects to be labeled, and 2) the set of features of the object that can be extracted by observing the object itself. Unlike traditional classification models that make independence assumptions and only model the features within each object, such as Naïve Bayes [16] and Maximum Entropy [17], sequence modeling methods exploit the dependence among the objects. Such methods include Hidden Markov Models (HMMs) [18], Maximum Entropy Markov Models (MEMMs) [19] and Conditional Random Fields (CRFs) [20]. HMMs, based on

a directed graphical model, have been widely used to label sequences. HMMs model the joint probability distribution $p(\mathbf{y}, \mathbf{x})$ where $\mathbf{x}$ represents the features of the objects we observed and $\mathbf{y}$ represents the classes or labels of $\mathbf{x}$ we wish to predict. MEMMs, also based on a directed graphical model, combine the idea of HMMs and Maximum Entropy (MaxEnt). (CRFs) [20] are based on an undirected graphical model. CRFs directly model the conditional distribution $p(\mathbf{y}|\mathbf{x})$. It follows the maximum entropy principle [21] shared by MaxEnt and MEMMs.

3.2 Sentence Classification

Sentence classification has been studied in previous work. Khoo, et al., evaluated various machine learning algorithms in an email-based help-desk corpus [22]. Zhou, et al. studied the multi-document biography summarization problem based on sentence classification [23]. However, in these two approaches, the sentences are treated independently from each other. No interdependencies were considered.

Jindal and Liu studied the problem of identifying comparative sentences in text documents [24]. Such sentences contain two or more objects and the comparisons between them. Their proposed approach uses a combination of class sequential rules(CSR) and machine learning. CSRs are based on sequential pattern mining, which finds all sequential patterns that satisfy a user-specified minimum support constraint. This makes CSRs fundamentally different from our sequential data labeling task.

Hachey and Grover evaluated a wide range of machine learning techniques for the task of predicting the rhetorical status of sentences in a corpus of legal judgements [25]. They examined classifiers making independence assumptions, such as Naïve Bayes and SVM. They also report results of a Maximum Entropy based model for sequence tagging [26]. This approach is similar to MEMMs. However, they evaluate only one sequence labeling model and the features for sentence classification are limited. We have identified a richer set of features that are effective for sentence classification in our domain of interest.

3.3 Destination Name Extraction

Our task of extracting the names of destinations is related to the field of named entity recognition (NER). NER aims to extract entities such as names of persons, organizations, or locations from text. Much work on NER has focused on machine learing methods, such as Maximum Entropy [17] [27], Hidden Markov Model [28], CRFs [20]. However, our task of finding destination names is fundamentally different from the traditional NER task. In driving directions, there are many names. However, most of them are landmarks in the instructions to help travelers locate themselves and make route decisions; only one or a few of them are names of the destination. Additionally, traditional NER methods suffer from the ungrammatical nature of Web pages, such as over-capitalization and under-capitalization [29].

4 Algorithm Description

In this section, we describe our proposed algorithms that classify the route components and identify the destination names in a direction document. Given the list of sentences

(in their original orders) extracted from a document containing route directions, we first use machine-learning algorithms to classify each sentence into one of the four route labels: "DESTINATION", "ORIGIN", "INSTRUCTION" and "OTHER". Based on the classification results, we apply our proposed algorithm to find candidates for destination names from two sources: (1) sentences predicted as a "DESTINATION", or (2) the "arrival information" sentences from the predicted results. Using these extracted candidates, we re-check the sentences labeled as "DESTINATION" and "OTHER" to pick up mis-classified true destinations and finalize the set of candidates.

4.1 Sentence Classification

We use four machine learning models for sentence classification: CRFs, MEMMs, Naïve Bayes, and Maximum Entropy. A list of sentences is extracted from route descriptions using the method described in our previous work [12]. Then, these models use the following features extracted from each sentence to perform the classification:

Bag-of-Words Features: The appearance of each term in a sentence is a feature of the sentence. The same term in different cases are considered to be the same (the case information is captured by the surficial feature discussed next). The algorithms do not eliminate traditional stop words because some of them play an important role in our classification task, for example, "take", "onto" and "at" carry important spatial clues and should not be eliminated; others that do not carry spatial clues do not impact the classifier.

Surficial Features: Surficial features describe the "shape" of the sentences. They are: whether a sentence has only one word, whether a sentence consists of digits only, whether all words have their initials capitalized, and whether all letters are capitalized. We designed this set of features to capture the way certain route components are expressed. For example, the destination and origin may have been emphasized in the document by capitalizing each letter (e.g.,"*DIRECTIONS TO HOSPITALITY HOUSE*", "*FROM THE NORTH*", etc.); phone numbers of the destination, which are all digits, should be labeled as "other" (note that phone numbers can be used to query online phone book services to help identify destinations, however, in this work, we do not consider external knowledge sources and only extract the features within each document).

HTML Visual Features: For direction documents from the Web, different route components have different HTML visual features. HTML authors often use different visual features for different route components. Titles of HTML documents may contain the destination; destinations and origins are often in Headings; links in HTML are often irrelevant to route components. Therefore, we extracted the information about whether a sentence is a title, a link or a heading, as a set of features.

Domain-specific Features: We identified a set of frequent patterns that appear in directions. Such patterns include highway names and particular verb phrases, such as "turn left to ...", "merge onto ..." and "proceed ... miles...". We refer to these patterns as **language patterns**. Using a set of rules consisting of regular expression patterns, we check whether a sentence contains any of these patterns. We designed the set of rules based on an examination of a sample of documents obtained from our collected corpus (Section 5.1 describes how the corpus was built). In our system, we have 24 regular expressions to extract frequent language patterns for instructions, 2 for destinations,

Table 1. Regular Expressions to extract domain-specific features

Feature	Regular Expressions	Example
DEST1	\s*(driving)?\s*(direction|directions)\s+to\s+\w{2,}.*	driving directions to IST
DEST2	.*visiting\s*\w{2,}.*	Visiting Winterthur
ORIG	.*(direction|directions|coming)?\s*(from—via)\s* HighwayPS? (the)?\s? CardinalDirPS .*	coming from I-80 East
INST1	.*turn\s+(left|right).*	turn left
INST2	.*turn \s+off\s+at\s+exit(\s+\d+)?.*	turn off at Exit 168
INST3	.*(proceed|travel|drive|go|turn|head)\s+CardinalDirPS.*	traval north for 3 miles.
INST4	.*(proceed|continue)\s+(along|into|past|on).*	proceed along College Avenue
INST5	.*continue\s+(CardinalDirPS\s+)?.*	continue east
INST6	.*take\s+(?:the\s+)?(?:OrdinalNum\s+)?(\w*\s+)?exit.*	take the second exit
INST7	.*follow\s+the\s+exit.*	follow the exit to State College
INST8	.*follow\s\d{1,5}(\.\d{1,5})?\smile(s)?.*	follow 3.4 miles
INST9	.*follow\s+(the\s+)?sign.*	follow the sign to San Jose
INST10	.*follow\s+(the\s+)?(\w+\s+)*(avenue|street|road).*	follow College Avenue
INST11	.*follow\s+ HighwayPS .*	follow I-80 west
INST12	.*take\s+ HighwayPS [./]*	take US-322/220
INST13	.*exit\s+(at\s+)? HighwayPS .*	exit at PA Ruote 23
INST14	.*exit\s+(left|right)?\s*(at|onto|to)\s+.*	exit right onto KING Rd
INST15	.*stay\s+(on|straight).*	stay on I-80
INST16	.*(make|take)\s(left|right)\s(onto|to|on).*	make left onto Columbus Blvd.
INST17	.*(make|take)\s+(the|a|(the\s+(next|CardinalDirPS))) \s+(\w*\s+)?(left|right).*	take a sharp left
INST18	.*(bear|keep)\s+(to\s+the\s+)?(left|right).*	bear left
INST19	.*merge\s+(left|right)?\s*onto.*	merge onto I-670 E
INST20	.*take.*(exit|ramp|bridge|turnpike|thruway|expressway).*	take NJ Turnpike South to
INST21	.*HighwayPS \s+CardinalDirPS\s+exit.*	to Route 322/22 West Exit
INST22	.*exit\s*\d+.*	take exit 168.
INST23	.*StreetTypesPS\s*CardinalDirPS.*	East 49th Street north
INST24	.*CardinalDirPS .* StreetTypesPS.*	...north Atherton Street...
OTHER1	.*\b[a-zA-Z0-9._%-]+@[A-Z0-9.-]+\.[A-Z]2,4\b	match email addresses
OTHER2	\W+	match lines with only non-word characters

1 for origin and 2 for other. Table 1 gives all the regular expressions and examples of language patterns in the text ("HighwayPS" is the pattern string for highway names; "CardinalDirPS" is the pattern string to match the orientation words such as east and west). Another set of domain-specific features are nouns and noun phrases frequently used to specify the types of destinations, such as "hotel", "restaurant", "campus", etc. We created a **dictionary** of such nouns and noun phrases referring to a place or a location. Whether a sentence contains a dictionary entry is a binary feature of this sentence.

Window Features: are specified language patterns that appear in a specific window around a sentence. Window features are extracted after the surficial and language pattern features have been extracted. Our system checks whether specified features or any one of a set of specified features exists in the previous and/or the following sentences of the current sentence. One window feature checks if the surrounding sentences match any of the 24 regular expressions for instructions. Since instructions are often grouped

together, the current sentence may also be an instruction if it has this window feature. Consider the following three consecutive sentences extracted from the document shown in Figure 1: (1) *"Head southwest on Fox Hill Rd. toward High Tech Rd. (2.4 mi)"*, (2) *"Continue on Fox Hollow Rd. (1.9 mi)"* and (3) *"Turn right at E. Park Ave. (1.5 mi)"*. When we examine the second sentence, we can see that sentence (1) and (2), match language pattern INST3 and INST1 (see Table 1), respectively. Therefore, sentence (2) has the window feature we described above. Other window features are: whether there is a street name and number in the previous/following two sentences, whether there is a zip code in the following two sentence.

4.2 Destination Name Recognition and Re-classification

Deficiencies in Sentence Classification Results. The machine-learning models give us the list of sentences with predicted route component labels. After analyzing the classification results, we found that the classification accuracy for destination sentences was not satisfactory (see Section 5.2). Although the best precision for destinations can reach about 80%, the recall is about 40% for all MEMMs and CRFs. Since destinations are a very important route component in descriptions of directions, the recognition of destinations from the list of sentences is crucial.

Based on our analysis, we noticed that many destination sentences were classified as "OTHER" by mistake (the reasons will be given in Section 5). Such destination sentences often contain the name of the destination. Their features are very similar to the features displayed by "OTHER" sentences. For example, the name of the business or institute may stand alone as an individual sentence and not have the strong features associated with destinations. Such sentences are often very short and do not have many language pattern features to extract. The terms in the name are often capitalized as they are in "OTHER" sentences. Sometimes abbreviations are used for the destination name, which makes it more similar to entries classified as "OTHER". Without looking for more information from the context, it will be very difficult to identify such destination sentences with destination names with high accuracy.

Luckily, when people create route descriptions, they sometimes put destination names or the type of the destination in the arrival information of a set of route instructions. The names are often embedded in an obvious language pattern. We show some examples of the mis-classified destination sentences and the contextual information we can utilize to improve classification below:

Example 1:
Destination: Trial Lawyer Harford Connecticut - Levy and Droney P. C.
Arrival Info.: Levy and Droney is located in the building on the right.
Example 2:
Destination: 2002 - 2005 Atlantic Country Club
Arrival Info.: Atlantic Country Club will be 2 miles down on the left.

Once the destination name has been identified from the arrival information, we can use the names to match the names in the sentences predicted as "OTHER" and re-classify them.

In addition, the destination names in the sentences classified as "DESTINATION" can also be used to find the mis-classified destination sentences. Given the high

precision of predicted destination sentences, it is reasonable to assume that most destination names can be found in the sentence already predicted as "DESTINATION". Once the names are extracted, we can again use them to identify mis-classified destinations. Here are some examples of such cases. In each one of them, the mis-classified sentence is a destination but classified as "other".

Example 3:
Correctly Identified: Delaware Court Healthcare Center - Map (Title)
Mis-classified: Delaware Court Healthcare Center (at the end of the document)
Example 4:
Correctly Identified: Directions to NIBIB (at the beginning)
Mis-classified: About NIBIB (at the end of the document)

In Example 3, since being in the title is a strong indicator of being a destination, the first sentence is classified correctly. However, the second one was not recognized since it is surrounded by "other". The dependency assumption caused the classifiers to make the wrong decision. The reasons are the same for Example 4.

Destination Name Extraction. We take the output of the sentence classifiers (a list of sentences with predicted route component labels for each one) and find blocks of instructions. Then the last sentence, or arrival information, of each block is extracted. Since destination names are proper nouns or proper noun phrases and destination types are a relatively small set of nouns, we rely on a Part-of-Speech tagger [2] and the dictionary of type nouns (introduced in Section 4.1) to identify the nouns and proper nouns in these sentences. However, not all proper nouns are destination names; we have to filter the results. Arrival information often has language patterns and the destination names are at certain positions of such pattern. Consider the sentence: "*The church is on your left.*". When the pattern "*... is on your left*" is identified, the nouns in the front of it are very likely to be the destination name or type, thus the range for searching for the destination names or types is narrowed from the entire sentence to a portion of it. We created a set of regular expressions to represent such patterns and specified the places where the destination names and types can be found in each regular expression. Our algorithm matches the arrival information sentences against these patterns. If it finds a match, the algorithm only focuses on the nouns and proper nouns in part of the sentence. If not matched, we use a heuristic rule to filter the results: the noun or noun phrase must have at least one proper noun or one type noun in it, otherwise it is discarded. After that, we further remove the nouns and noun phrases that are able to pass the above filtering steps and are preserved, but are not destination names such as "*parking lot*"s or "*parking garage*"s without names. We remove them from the returned results. Algorithm 1 gives the details. The same algorithm can be also applied to the sentences predicted as "destination" to extract the names or types from them, except that a different set of regular expressions are used. In our actual system, both "DESTINATION"s and "ARRIVAL INFORMATION" are used as the input to this algorithm and it returns a set of candidate destination names and types.

Improving Recognition of Destinations. After extracting possible destination names and types from arrival information and predicted destination sentences, we use them

[2] In our system, we used LingPipe POS tagger: http://alias-i.com/lingpipe/

Algorithm 1. Extract Possible Destinations from Arrival Information

Input:
(1)a sentence s (2)a trained POS tagger T (3)a dictionary of type nouns D (4)a set of patterns P
Output:
a set of possible destination names $Dest$
Procedure:

```
1:  Dest ← ∅;
2:  find all nouns and noun phrases in s using T;
3:  if s matches any pattern in P then
4:      find all nouns and noun phrases in the capturing group and insert into n;
5:  else
6:      for each noun or noun phrase n in s do
7:          if n has at least one type noun in D or a proper noun then
8:              insert n into Dest;
9:          end if
10:     end for
11: end if
12: remove fake destinations from Dest;
13: return Dest;
```

to re-examine the sentence predicted as "OTHER". During the re-examination, we also filter the extracted destination names and types to generate the final set of possible destination names and types to return to the users. We first pick all the type nouns in the set generated from Algorithm 1 and search for them in the sentences labeled as "DESTINATION" and "OTHER". If this type noun appears in the sentence, we find the proper nouns in front of this type noun and consider the whole noun phrase as one of destination names. Then we change the label to "DESTINATION". After processing the type nouns, we match proper names against the sentences. Again, if this proper name is found in the sentence, we consider it as one of the destination names and change the label to "DESTINATION". At the end, the re-classified sentences and the set of destination types and names will be returned to the user (see Algorithm 2).

5 Experiment Results

In this section, first, we describe how we built our data set. Then, we give the sentence classification results for different machine-learning models. The results for destination-name extraction and re-classification are reported afterwards.

5.1 Data Set

We identified a set of over 11,000 web pages containing route directions using the search results of the Yahoo![3] search engine. The search engine was queried with a set of carefully selected keywords such as "direction, turn, mile, go", "turn, mile, follow, take, exit" etc. since these phrases are typically present within documents containing

[3] www.search.yahoo.com

Algorithm 2. Re-classification and Destination Name Extraction

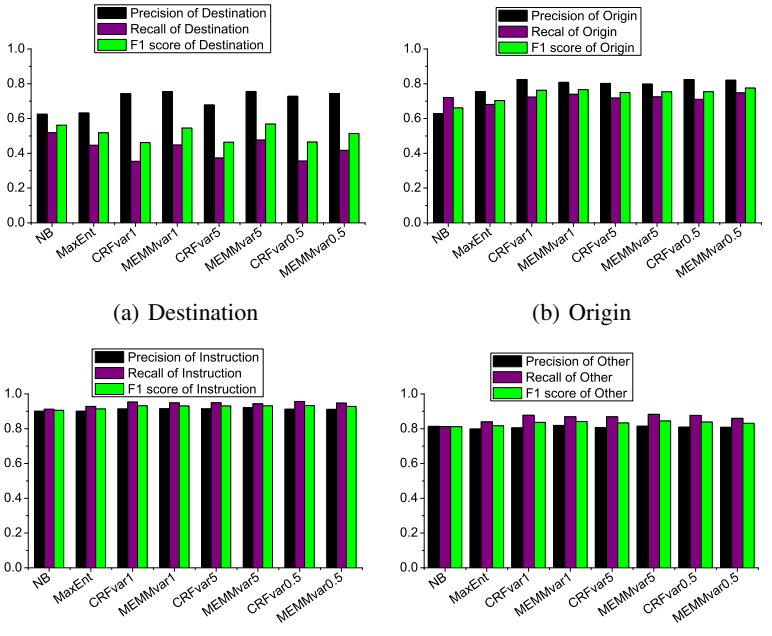
Input:
(1) L: a list of sentences with predicted route component labels
(2) S_{cand}: a set of proper names or types extracted from predicted destinations and arrival information.
Output:
(1) L': a new list of sentences with route components labels
(2) S_{dest}: a set of destination names or type
Procedure:

```
1: S_dest ← ∅, L' ← L;
2: for each sentence l labeled as "destination" or "other" in L' do
3:    for each type noun t in S_cand do
4:       if l contains type noun t then
5:          extract the proper name n in front of t in l, insert n into S_dest;
6:          change the label of l to be "destination";
7:       end if
8:    end for
9:    for each proper name p in S_cand do
10:      if l contains p as a substring then
11:         insert p into S_dest;
12:         change the label of l to be "destination";
13:      end if
14:   end for
15: end for
16: for each sentence l labeled as "destination" in L' do
17:    extract proper names and type nouns in l and insert into S_dest;
18: end for
19: return L' and S_dest;
```

route directions. Each set of queries contains 4 to 7 keywords. Manual examination shows 96% of these documents contain route directions.

5.2 Sentence Classification Results

We evaluated four machine-learning models. Two of them are sequence-labeling models (CRFs and MEMMs). In order to examine the sequential dependency of route components, we use two other models based on the independence assumption: Naïve Bayes and Maximum Entropy models. For CRFs and MEMMs, we tried different values for the initial Gaussian Variance: 1.0, 5.0 and 0.5. The set of sentences we used to train and test the models contains over 10,000 hand-labeled sentences extracted from 100 HTML documents. We used a 10-fold cross validation technique to evaluate these models. Note that in order to preserve the sequential natural of the sentences, we divided this data set by documents, not by individual sentences. As shown in Figure 2, MEMMs and CRFs outperform Naïve Bayes and Maximum Entropy. The recognition of "Origin", "Instructions" and "Other" are reasonable. However, recognizing "Destination" is a hard problem. "Destination" sentences are often very short and lack effective features. This is the reason we utilized the arrival information to find the destination names.



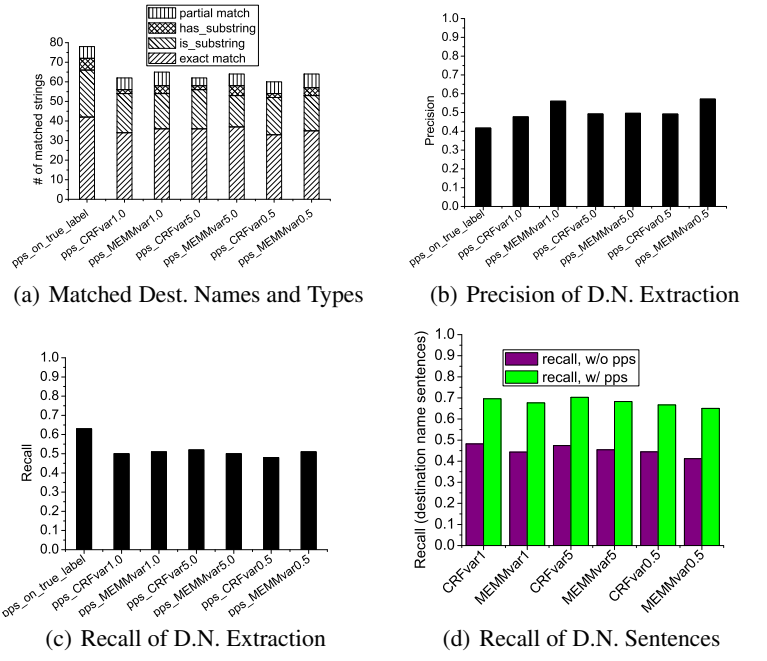
(a) Destination (b) Origin

(c) Instruction (d) Other

Fig. 2. Detailed Analysis of Each Class

5.3 Destination Name Extraction

We applied our proposed algorithms as a post-processing step (PPS) after we obtained the sentence classification results. First, we evaluate the performance of destination name and type extraction. We picked 46 of the 100 documents and hand-labeled a total of 100 destination names and types. For example, in one document, the destination is called “Berkeley campus”. It is also referred to as “campus” or “university” in other parts of the document. All three are hand-tagged as destinations. This set of destination names is our ground truth. PPS is applied to the CRFs and MEMMs classified sentences, as well as hand-labeled sentences, to extract destination names and types. We do so to evaluate how well the PPS performs without the noise introduced by the classifiers. We consider the following matching schema between the extracted names and types: the extracted name (1) has an exact match in the ground truth; (2) is a substring of an entry in the ground truth; (3) contains an entry in the ground truth as a substring; (4) is a partial match (they share common terms but do not have substring relationship); or (5) does not match at all. Figure 3(a) shows the number of names/types in different matching schema. As we have expected, PPS on true labels yields the highest number of exact matches. PPS on MEMMs with variance 5.0 and 1.0 gives the best exact matches and total matches among different classifiers. The precision is the number of matched names divided by the total number of names retrieved by the PPS. Recall is the number of correctly retrieved **unique** names and types in the ground truth divided by the total number of names in the ground truth. Note that when calculating recall, the same name or type that appeared multiple times is counted only once. Due to this reason, precision



(a) Matched Dest. Names and Types (b) Precision of D.N. Extraction

(c) Recall of D.N. Extraction (d) Recall of D.N. Sentences

Fig. 3. Extraction of Destination Names and Types

and recall are *NOT* calculated on the same objects. Therefore, no F1 score (a weighted average of the precision and recall) is given here. Figure 3(b) and 3(c) gives the details.

5.4 Recognition of Sentences with Destination Names

The extracted destination names and types from the classification results are then used to re-classify the sentences labeled as "Other". Figure 3(d) shows that the recall of sentences containing destination names or types has been improved substantially in comparison to the classification results without PPS; this proves the effectiveness of the PPS module. Note that in the destination sentences, there are also addresses or parts of addresses. We have not counted them in the calculation of the recall. Calculating the precision of destination name sentences requires counting the false positives (sentences labeled as "destination" by mistake). However, this number is hard to get because it is not easy to tell which mistake is introduced by the addresses.

5.5 Further Discussions

We also analyzed the destination sentences that are not predicted correctly and not recognized by the post-processing algorithm. We found that some destination names are stand-alone sentences without strong features and thus mis-classified. What made the problem even harder is that, in the arrival information the author of the route description used another name for the same destination. Therefore, they are not picked up in the post-processing step. Here is an example: at the beginning of the document, the

destination is mentioned as “Delaware Academy of Medicine”. The phrase itself is a “sentence” (for our purposes). In the arrival information, the destination is mentioned as “Christiana Hospital campus”. Our post-processing algorithm successfully recognized this phrase and considered it as a candidate for destination names. However, linking the two names together requires external knowledge. We do not have the ability to do such linking now, but will integrate external knowledge, such as that obtained from a geographic database, into the system in the future.

6 Conclusions and Future Work

All movement patterns of individual entities consist of three distinct parts: origins, destinations, and the path/route. The importance of understanding and identifying these parts has been recognized in different disciplines and has a central place in interpreting movement patterns from a cognitively relevant spatial perspective. While quantitative approaches are needed to handle the avalanche of data that becomes available through the development of spatial (tracking) technologies, we focus on the equally important problem of developing tools and methods to handle the qualitative data avalanche. Web documents or transcriptions of verbal communications are some of the most important sources of spatial information, and spatial language is one of the most challenging to handle automatically. We discussed our work toward automatic extraction of route components from documents containing route directions. We focus primarily on the extraction of destination names and types and on improving the recognition of destination sentences. Our system uses trained machine-learning models to classify sentences into route components. Based on the results of this classification, we proposed new algorithms that extract possible destination names and types from certain parts of the document. With this new knowledge, the system checks the re-occurrence of the extracted names and types to find misclassified destinations so that the recall is improved. Experiments have shown the effectiveness of our new methods.

A future step is to query various external information sources, such as online phone book services, web search engines and digital maps, and combine the returned results, in order to identify and geo-code the destinations. Another challenge is the occurrence of multiple destinations and origins that may appear in one document and that are sometimes not well ordered. Thus, finding the correct association of destinations, origins and instructions to form a complete route will be our next step. Additionally, we will also work on matching direction descriptions to GIS databases and geographic ontologies to support both disambiguation and enable human interpretation and refinement.

Acknowledgement

Research for this paper is based upon work supported National Geospatial-Intelligence Agency/NGA through the NGA University Research Initiative Program/NURI program. The views, opinions, and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Geospatial-Intelligence Agency, or the U.S. Government.

References

[1] Mark, D.M., Frank, A.U. (eds.): Cognitive and linguistic aspects of geographic space. Kluwer, Dodrecht (1991)

[2] Lakoff, G., Johnson, M.: Metaphors we live by. University of Chicago Press, Chicago (1980)

[3] Johnson, M.: The body in the mind: The bodily basis of meaning, imagination, and reasoning. University of Chicago Press, Chicago (1987)

[4] Allen, G.: Principles and practices for communicating route knowledge. Applied Cognitive Psychology 14(4), 333–359 (2000)

[5] Pick, H.: Human navigation. In: Wilson, R.A., Keil, F.C. (eds.) The MIT encyclopedia of the cognitive sciences, pp. 380–382. MIT Press, Cambridge (1999)

[6] Talmy, L.: Fictive motion in language and "ception". In: Bloom, P., Peterson, M.P., Nadel, L., Garrett, M.F. (eds.) Language and space, pp. 211–276. MIT Press, Cambridge (1996)

[7] Kurata, Y.: The 9+-intersection: A universal framework for modeling topological relations. In: Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2008. LNCS, vol. 5266, pp. 181–198. Springer, Heidelberg (2008)

[8] Egenhofer, M.J., Herring, J.R.: Categorizing binary topological relations between regions, lines, and points in geographic databases: Technical Report, Department of Surveying Engineering, Univeristy of Main (1990)

[9] Kurata, Y., Egenhofer, M.J.: Interpretation of behaviors from a viewpoint of topology. In: Gottfried, B., Aghajan, H. (eds.) Behaviour monitoring and interpretation. Ambient intelligence and smart environments, pp. 75–97 (2009)

[10] http://www.cost.esf.org/domains_actions/ict/Actions/IC0903-Knowledge-Discovery-from-Moving-Objects-MOVE-End-date-October-2013

[11] Miller, H.J.: The data avalanche is here. Shouldn't we be digging? Journal of Regional Science (in press)

[12] Zhang, X., Mitra, P., Xu, S., Jaiswal, A.R., Klippel, A., MacEachren, A.: Extracting Route Directions from Web Pages. In: WebDB 2009 (2009)

[13] Golledge, R.G.: Human wayfinding and cognitive maps. In: Golledge, R.G. (ed.) Wayfinding behavior. Cognitive mapping and other spatial processes, pp. 5–45 (1999)

[14] Denis, M., Pazzaglia, F., Cornoldi, C., Bertolo, L.: Spatial discourse and navigation: An analysis of route directions in the city of Venice. Applied Cognitive Psychology (1999)

[15] Manning, C.D., Raghavan, P., Schüze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)

[16] Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 4–15. Springer, Heidelberg (1998)

[17] Borthwick, A.: A maximum entropy approach to named entity recognition. Ph.D. thesis, New York University (1999)

[18] Freitag, D., McCallum, A.: Information extraction using hmms and shrinkage. In: AAAI Workshop on Machine Learning for Information Extraction (1999)

[19] McCallum, A., Freitag, D., Pereira, F.: Maximum entropy markov modes for information extraction and segmentation. In: Proceedings of ICML (2000)

[20] Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML (2001)

[21] Berger, A.L., Pietra, S.A.D., Pietra, V.J.D.: A maximum entropy approach to natural language processing. In: Computational Linguistics (1996)

[22] Khoo, A., Marom, Y., Albrecht, D.: Experiments with sentence classification. In: ALTW (2006)

[23] Zhou, L., Ticrea, M., Hovy, E.: Multi-document biography summarization. In: Proceedings of EMNLP (2004)
[24] Jindal, N., Liu, B.: Identifying comparative sentences in text documents. In: Proceedings of SIGIR, pp. 244–251 (2006)
[25] Hachey, B., Grover, C.: Sequence modelling for sentence classification in a legal summarisation system. In: Proceedings of 2005 ACM Symposium on Applied Computing (2005)
[26] Ratnaparkhi, A.: A maximum entropy part-of-speech tagger. In: EMNLP (1996)
[27] Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named Entity Recognition with Character-level models. In: CoNLL-2003, pp. 180–183 (2003)
[28] Bikel, D.M., Schwartz, R.L., Weischedel, R.M.: An algorithm that learns what's in a name. Machine Learing 34(1-3), 211–231 (1999)
[29] Ding, X., Liu, B., Zhang, L.: Entity Discovery and Assignment for Opinion Minig Applications. In: KDD 2009 (2009)

Visual Exploration of Eye Movement Data Using the Space-Time-Cube

Xia Li[1,2], Arzu Çöltekin[3], and Menno-Jan Kraak[1]

[1] ITC- University of Twente,
P.O. Box 6, 7500 AA, Enschede, The Netherlands
{xia,kraak}@itc.nl
[2] College of the Earth Science and Resource–Chang'an University,
126# Yanta Road, Xi'an, China
[3] Department of Geography, University of Zurich,
Winterthurerstr. 190 CH-8057 Zurich, Switzerland
arzu.coltekin@geo.uzh.ch

Abstract. Eye movement recordings produce large quantities of spatio-temporal data, and are more and more frequently used as an aid to gain further insight into human thinking in usability studies in GIScience domain among others. After reviewing some common visualization methods for eye movement data, the limitations of these methods are discussed. This paper proposes an approach that enables the use of the Space-Time-Cube (STC) for representation of eye movement recordings. Via interactive functions in the STC, spatio-temporal patterns in eye movement data could be analyzed. A case study is presented according to proposed solutions for eye movement data analysis. Finally, the advantages and limitations of using the STC to visually analyze eye movement recordings are summarized and discussed.

Keywords: Eye movement analysis, Space-Time-Cube, Usability evaluation, Spatio-temporal data.

1 Introduction

Usability evaluations of visual repsentations have been drawing much attention in recent GIScience and visual analytics research [1, 2, 3, 4, 5]. The evaluations typically deal with user requirements, trying to find out how people solve spatial problems and what cognitive processes might be behind their actions. To be able to derive qualitative or quantitative measures of the user experience, a number of evaluation methods have been tested. Some examples for such methods can be listed as: focus group studies, interviews, direct observations, think-aloud protocols, retrospective think-aloud protocols, screen logging and eye movement recording and analysis. Recoding eye movements does not rely on self-reporting, therefore it can be considered an objective method and can enhance traditional performance tests, protocol analysis, and walk-through evaluations of a system [6].

Eye tracking research results in an enormous amount of highly detailed data. Typically a time stamp (temporal data) and gaze point location within the configured

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 295–309, 2010.

screen coordinate system (spatial data) is reported by the tracker. One of the challenges, as tackled by many researchers [2, 6, 7, 8, 9, 10] is how to process, manage, and use these continuous streams of data efficiently and effectively to support a usability evaluation.

Within the scope of this paper, the following questions are most relevant: Is it possible to analyze eye movement data using traditional spatio-temporal tools that have been used in spatial analysis domain? Can geo-visual analytics methods be used to improve the detection and comparison of possible spatio-temporal patterns in eye movement data? How can these methods be combined with typical eye movement analysis methods, such as gaze plots, density maps, AOI (area of interest) analysis and statistics? Do above ideas provide further insight into understanding and interpreting eye movement data? With above questions in mind, this paper tests an approach that combines typical eye movement analysis methods and the Space-Time-Cube (STC), which has been used in geography since its introduction by Hägerstrand in 1970 [11].

2 Eye Movement Data Analysis and State-of-the-Art in Spatio-Temporal Geovisualization

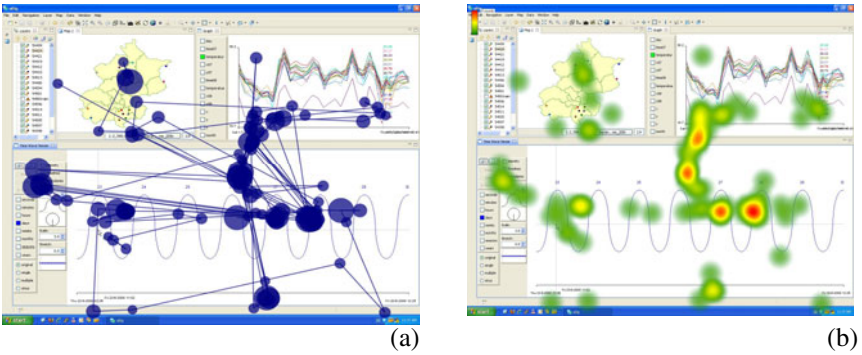
Along with gradually maturing hardware technology to track eye movements, the applications (software) that utilize information derived from eye movements is also becoming more and more comprehensive. Eye movement recording and analysis may offer additional tools to enhance usability studies. Several research papers integrating usability studies and eye movement analysis have also been published in geovisualization domain [1, 2, 3, 12, 13] and appear to continue attracting attention. While clearly useful, processing, analyzing and interpreting data that is collected via eye tracking is still cumbersome and arduous. Can some developments in subfields of geo-visual analytics, such as the dynamic, interactive, 3D STC be helpful in making this easier? With this question in mind, some common methods for eye movement data processing and analysis, as well as some current spatio-temporal geovisualization methods will be discussed in the following sections.

2.1 Eye Movement Data Analysis

Eye movement studies have been conducted long before computers were widely used. According to Jacob [14] such studies for basic psychological research existed already about 100 years ago, addressing a set of versatile questions; *e.g.,* it has been used when studying language comprehension and production [15], scene perception [15, 16], reading [17] or spatial reasoning [18]. Eye movement analysis has also been introduced and integrated into usability studies. Goldberg and Kotval [6] contend that performance and usability evaulations of spatial displays within information acquisition contexts, eye movement analysis has at least a 70 year long history. In geovisualization, using eye movement recording and analysis for evaluating the user performance can be considered both ‘old’ and ‘new’. In this context, ‘old’ means that many typical evaluation studies supported by eye movement recording and analysis in other disciplines could also be used for geovisualization. Such studies exist, for

example, in computer interface evaluation [6, 19, 20, 21, 22], human computer interaction (HCI) usability testing [7, 23, 24, 25] and cognitive processing [20, 26, 27]. Early eye movement studies in geovisualization were driven by cartographic research questions [3, 28]. 'New' refers to developments in technology that enable us to collect more data than ever before and new methods in geovisual analytics which enable us to design and visualize complex and dynamic processes. Eye tracking offers new promises and challenges in evaluating and analyzing the cognitive processes when people use these large, complex, often interactive data visualizations. However, eye movement data itself is typically very large.

To analyze and make use of the huge amounts of collected eye movement data efficiently, abundant research has been done (e.g., [8, 9, 10, 29, 30, 31, 32]). When dealing with traditional eye tracking data, several metrics are reported in usability studies, such as: fixation duration, gaze duration, area of interest (AOI) analysis, and scan path comparisons. These metrics are used to analyze the visual search processes of users as well as to establish the location of their overt attention. For instance, such metrics can be helpful to find out which part of the map attracts most attention at first glance, or what is the order of user gaze points while observing the map or solving tasks with it. These metrics can be represented as density maps, gaze plots and graphs. Fig. 1 shows an example of some common visualization methods used in representing eye movement data. On the left in Fig. 1, a gaze plot is shown which represents saccades, fixations and fixation durations plotted as a scan path. On the right, a density map can be seen which shows the average fixation duration of multiple users (density maps can also represent fixation counts). These representations provide a simple and direct static view of eye movement data and as such, they are included in most common eye tracking and analysis software.



(a) (b)

Fig. 1. Two common visualization methods used in visualizing eye movement data (screen view as background) produced in Tobii Studio. Left image shows a gaze plot, and the right image shows a fixation density map.

However, since eye movement datasets are almost always quite large, limitations of above visualization methods are obvious. Overlaps may cause misunderstanding of the data. For example, if viewer tries to identify the number of gaze points within a certain area this may not be possible, because larger fixations may occlude the smaller ones entirely. Furthermore, temporal information (such as the order of fixations on a

scan path) is potentially lost at overlapping scan paths and this makes it difficult to establish when the fixation is directed to a certain area. For smaller data sets, design choices such as using transparency, or numbering the fixation points can partly solve the problem, but in large sets of data this approach may not be feasible.

When used as stimuli, highly interactive, multiple-link-view environments in common geovisualization software pose additional problems. Scrolling windows, pop-up dialogs, animated graphics, user-initiated object movements, and other navigation features leave the experimenter with technical challenges for studying and interpreting fixations [14]. Existing common solutions for these problems in most eye tracking and analysis software consist of updating the viewed stimuli according to the screen view, showing mouse clicks on the interface and allowing animation. However, to analyze the data in a linear, continuous order both in temporal and spatial context, the above solutions are not always sufficient. One example of such a problem is to identify and compare spatio-temporal patterns 'created' by differences among users. Another example relates to differences between the hypothetical and real situations. Can spatio-temporal geovisual analytical methods help solve these problems?

2.2 State-of-the-Art in Spatio-Temporal Geovisualization

There is abundant literature discussing spatio-temporal data visualization. Most solutions are based on three cartographic depiction modes: a single static map, multiple static maps (*i.e.*, "small multiple maps") and a map animation [33]. A single static map is the 'simplest' visualization solution for spatio-temporal data and most users are familiar with it. However, it is difficult to represent complex changes, as depicted in Fig. 1a where gaze points of different time stamps overlap with each other. Small multiple maps represent the temporal sequence by a spatial sequence of individual maps each representing a moment in time. This facilitates to find the difference between any two time points of interest. However, it is also a discrete representation of the dynamic process. The number of images is limited, so it is difficult to deal with long series. With map animation, users might catch the "trend of change" more easily. A user can control the speed of the animation, and "stop" at any moment in time. However, it is also easy for the viewer to neglect the actual time point that the change happened, and the user might have difficulty to fix attention on multiple changing items [1].

An alternative visual representation is the Space-Time-Cube (STC), which is the most prominent element in Hägerstrand's space-time model [34]. The STC combines time and space in a natural way. Time can be represented as continuous or discrete. The X and Y axes indicate the 2D space, while the time units along the Z-axis can be years, days, hours, etc. In the STC, the Space-Time-Path shows the object's trajectory through time and space.

The recent revival in the interest in the STC is due to the development of new technologies which makes data collection as well as creation and use of the cube much easier than before [35, 36, 37, 38, 39, 40, 41, 42]. Typical geovisualization characteristics like interaction, dynamic and alternative views are now also applicable to the STC, and made it part of an exploratory environment. Andrienko [36] and Gatalsy [39] linked the STC with dynamic map displays by simultaneous highlighting corresponding symbols, and applied temporal focusing with the STC. With support of

rotating, panning, zooming in/out and similar functions, Johnson [43] and Forer [44] use the STC to display different snapshots along the time axis in a three-dimensional (3D) environment. Kraak [40] linked the STC with a 2D map view, video and attribute table to explore multi-variable, multi-media spatio-temporal data. Kraak and Madzudo [45] linked the STC with attribute graphs, such as bar charts and parallel coordinate plots (PCP) to interact with attribute patterns from both spatial and temporal perspectives. Another additional useful function is the option to move the 'base' map up and down along the time axis, which allows users to explore spatial distribution over time. Kraak [46] developed the STC in geovisualization environments further by intergrating Shneiderman's [47] viusal information seeking mantra (overview, zoom and filter, detail on demand) with the elementary spatial questions *where*, *what* and *when*. Another additonal function is the option to create paths with annotations [46]. Qualitative and quantitative information can be added as geo-tags to a path. This can supply extra multi-media detail within spatio-temporal context, such as pictures, videos, graphs, etc. and reduce the clutter in the STC view.

These progresses in research and development resulted in extending the functionality of the STC beyond its original design, and may supply more efficient ways to fully explore spatio-temporal data. Since eye movement data has both spatial and temporal characteristics, the next section will discuss whether it is possible to use the STC in eye movement data analysis and whether this helps solving some of the occlusion related problems with gaze plots and density maps.

3 Exploring Eye Movement Data in a STC Environment

A common approach to support problem solving with visualization is based on a combination of user tasks, a data framework and a visualization framework. With these 'constraints' in mind the next section will first discuss how the eye movement data can be represented by the STC, followed by an argumentation on how these representations can contribute to the understanding of the data, and help overcome some of the problems mentioned in the previous sections.

3.1 Eye Movement Data and Spatio-Temporal Data Modeling

Peuquet [48] distinguished three elements in spatio-temporal data: location, attribute and time (Fig. 2a). This data approach is widely accepted in geovisualization research. Eye movement data has similar characteristics, therefore it can be structured accordingly (Fig. 2a). The record's time stamp (or start time) for one gaze point corresponds to the time component in Peuquet's model; the X and Y of a gaze point (screen coordinates) represent the location component, and attributes could be for example validity, event data, gaze point content or AOI metadata. Hence, eye movement recordings have many similirties with spatio-temporal data and can be visualized as such. In an STC, the X, Y plane of the cube represents the user's screen view. The eye movement (Space-Time- Path) is along the Z axis. The movement's attributes can, for instance, be represented by the color, size (volume) of the path.

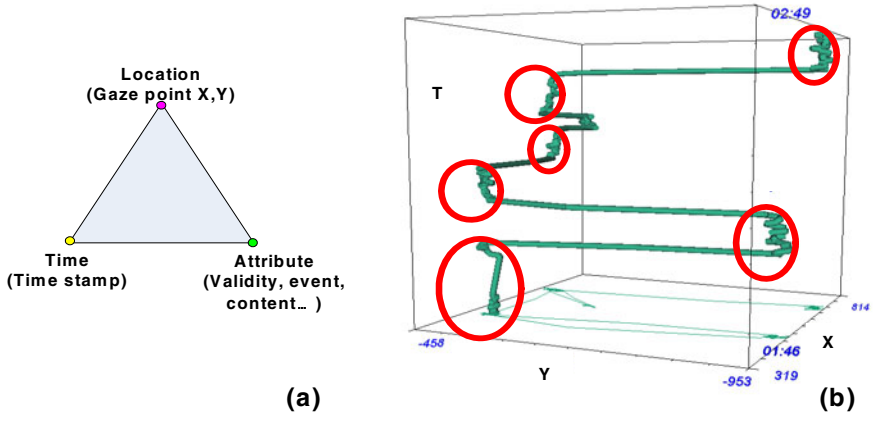


Fig. 2. (a) Eye movement data classified according to Peuquet's spatio-temporal model [48]; (b) a simple example of eye movement data in the STC.

Fig. 2b shows a simple example of eye movement data visualized in the STC. The trajectory of eye movements is displayed as Space-Time-Path (STP). It immediately reveals spatio-temporal patterns. The vertical 'lines' indicate (the ellipses in Fig. 2b) an eye fixation at a particular location. The fixations still include micro-movements as see in ellipses. This is not a surprise, because human eyes have continuous micro-saccadic movements and fixations are typically defined with temporal as well as spatial thresholds. In the STC, the fixation points can be easily identified by the approximate vertical line. The length of the approximate vertical line shows the duration of one fixation (fixation length). The horizontal lines indicate movements of eye (saccades). The slope of the line shows the speed of the eye movement. The Space-Time-Path can be projected on to a two-dimensional surface (screen view as background of the eye movement path), resulting in the familiar gaze plot representation with a scan path.

3.2 Solutions for Visual Analysis of Eye Movement Data with STC

In comparison with visualizations presented in Section 2.1, the STC shows spatio-temporal patterns of eye movement data in a three-dimensional view equipped with dynamic and interactive functions. One advantage of the three dimensional view is that, overlaps (as shown in Fig.1 and Fig. 3a) can be avoided. In addition, the temporal order of the eye movement is revealed (*e.g.,* compare scan paths in Fig. 3a and 3c) and various spatio-temporal patterns in eye movements of multiple participants can be identified and compared.

Several functions supported by geovisualization could further extend the power of the STC for exploration of eye movement data. A moveable base map (screen view) along the T-axis could help discovering to what the participants were attending at a certain moment (Fig. 4a). Another flexible function of the STC is applying "the visual information seeking mantra" [47]. 'Flexible' here means that filter and zoom functions work on attribute, time and location in a flexible way. The attribute filter could help users to select the additional interesting attributes, for example, the validity of

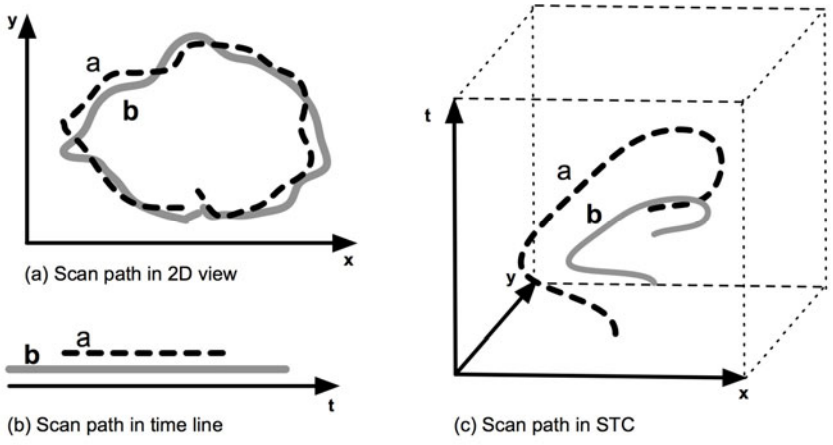


Fig. 3. Comparison of eye movements of two participants from different perspectives: (a) a 2D view shows the spatial pattern (potential overlap and no temporal order); (b) A timeline gives the temporal pattern only (no spatial information); (c) the STC reveals both spatial and temporal patterns.

gaze point data, or the records of one participant for one particular task. Both the time instant and interval could be visualized, supported by a temporal zoom function (*i.e.,* zoom time in Fig. 4b). For example, a segment or a scene from a screen recording (which mark the steps of a task) could be defined this way. The zoom function could work on location as well (zoom location in Fig. 4b). Using this function, spatio-temporal patterns of multiple participants on a certain AOI could also be compared. Furthermore, it is possible to define the spatio-temporal zoom in the overview of the STC. For example, in Fig. 4c, the spatio-temporal behaviour of Participant 1 and Participant 2 over the same AOI could be compared.

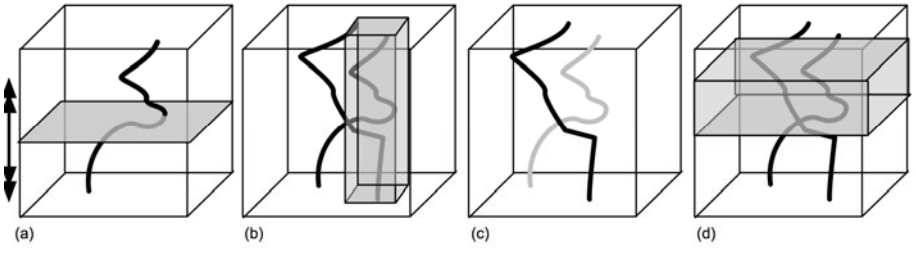


Fig. 4. Analytical functions in the STC (a) A moveable base map (screen view) along T-axis; (b) Spatial zoom; (c) Attribute filter; (d) Temporal zoom

Fig. 5a shows another possible function in the STC *i.e.,* the annotations path. Additional multi-media information, such as video and images, or statistical results and notes (*e.g.,* interviews results) of the individual participants could be attached as annotations to the STP. During the analysis phase, annotations could be retrieved to

access the detailed information with spatio-temporal content to further understand a participant's behavior. Fig. 5b and 5c shows a solution for analyzing the user behavior over a dynamic stimulus. In a geographic software environment, most frequently used functions are probably the spatial zoom in and zoom out. Understanding the geographic context is often very important for executing geographic tasks. In this case, the screen view after the spatial zooming could be projected back to the original scale. The x,y plan view adapts itself to the screen seen by the user. The time of the zooming operation could be shown with the Z axis. At this point, the scan path of the user could be reverted to the original scale, *i.e.,* overview scale, on the footprint map. Both spatial and temporal information of the zooming operations could be displayed. If dynamic view results in the non-geographic operations of a participant (*i.e.,* base map changes entirely) an optional solution could be showing the screen view along with the Z-axis or with STP.

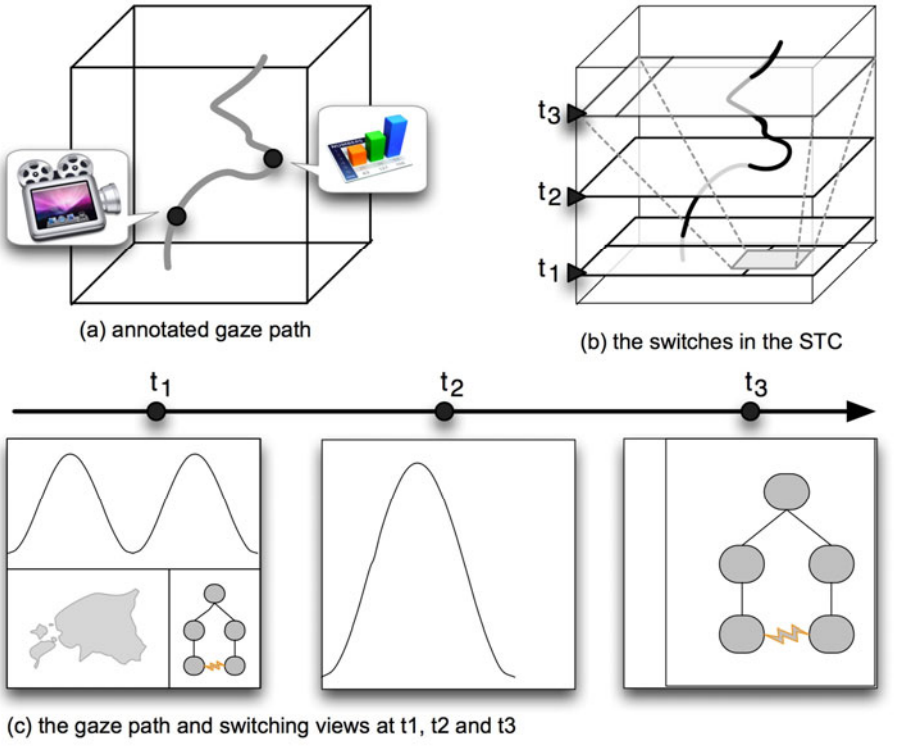


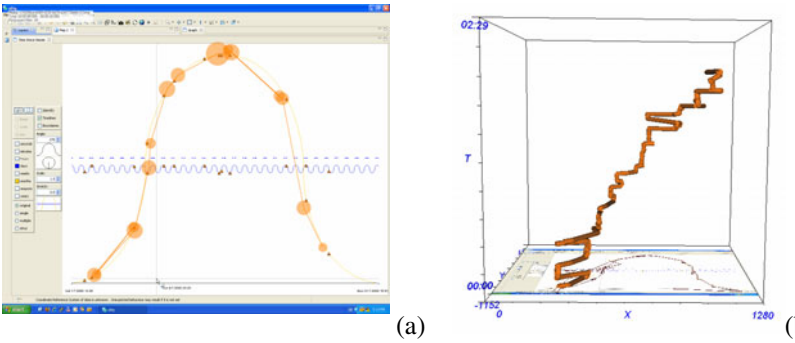
Fig. 5. Tracking the gaze path over a dynamic screen: (a) paths with annotations in times t1, t2 and t3; (b) user looks at overview screen (t1), zoom in on time wave (t2), and pan to graph (t3); (c) gaze path shown in the STC displaying the different screen views from (a) with the option to project them back into the original view.

4 Case Study Combining Eye Movement Data and the STC

The eye movement data in this case study was collected for evaluating an experimental geovisual analytics system. The interface of the system is based on the coordinated multiple view approach and includes a 2D map, a line, graph and a representation for temporal data and the time wave [49]. The objective of the evaluation was to judge user's behavior in this environment and get insight into users' thought processes while working with the time wave. In this paper, the eye movement data collected for the evaluation study is used independently from the experiment's original goals. Here it serves as a test dataset to investigate whether a geovisual representation, namely the STC, can be used to improve the understanding of patterns observed in the data. Data collection was done in a controlled laboratory setting (GIVA's Eye Movement Lab [50]), that is equipped with an active, near-infrared enabled remote video eye tracker (Tobii X120). In this study, the tracker was configured to record at 60 Hz sampling rate. The fixation threshold value was set to 100 milliseconds. Screen resolution was set to 1280*1024 pixels and the system was calibrated for each participant. The data post-processing stage involved creating scenes, segments, and AOI visualizations. To create an STP of eye movements in the STC, the attributes TimeStamp, GazePointX and GazePointY were used.

The core of the geovisual analytcis system consists of uDig, an open source GIS software [51], with several dedicated plug-ins developed in-house such as, the time-wave and the STC. The STC functionality, which is most relevant in the scope of this paper, has been described in Section 3. Is it possible to use eye movement data to create STPs to visualize and qualitatively discover spatio-temporal patterns? Fig. 6 shows a comparison of a 'traditional' eye movement track represented in a gaze plot map overlaying a screen showing the time-wave (6a) and the same data in an STC. In Fig. 6a, it can be observed that the user's eyes followed the wave and stopped for a moment at the triangle points on the wave. These are the gaze points on the triangle and the scan path along the wave. However, the temporal order of gaze points cannot be determined in detail in this static image. Some scan paths that overlap can easily be misunderstood. In the STC view, as depicted in Fig. 6b, the trajectory does not follow the time wave straightforwardly, since it goes back and forth in places. This pattern cannot be derived from the 2D gaze plot map. The spatio-temporal pattern in the eye movement path is readily visible in the STC. The fixations can be identified by the approximate vertical section of the path. Thus, the STC visualization informs the viewer about the *when* and *how long* of the gaze behavior. In the experimental viewing environment, the STC is not the only available view; a 2D map is linked to the cube which gives the user the opportunity to follow the path in the cube and on the map at the same time, keeping control of the context view.

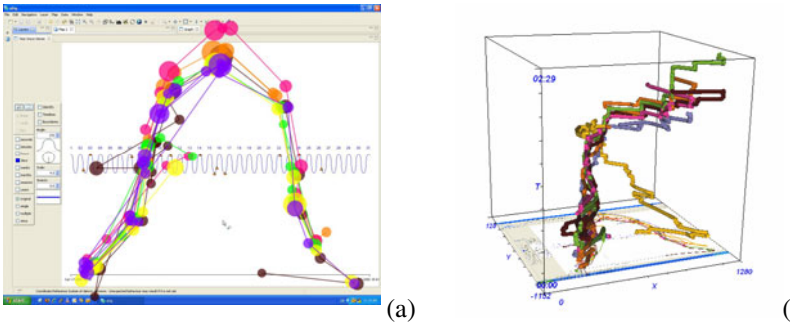
In Fig. 7, eye movement data for one particular task from a number of different participants is represented in the gaze plot and in the STC views. In the gaze plot all paths seem to be similar (Fig. 7a).



(a) (b)

Fig. 6. A comparison of a 'traditional' gaze plot representation of an eye movement recording (Tobii Studio) (a) with the same data in a STC (b)

From the STC (Fig. 7b) it can be concluded that this is not the case. One of the paths is clearly different from the others. This participant follows the wave right to left, while all the others went from left to right. The difference between participants' spatio-temporal patterns can be more easily distinguished in the STC visualization in comparison to 2D view where the 2D plot does not reveal the "odd case" as easily. The STC, in this case, offers a better visualization of spatio-temporal patterns, allowing a quick overview of the data, and in more detailed analysis. The STC can be combined with the other graphic representations with additional linked views.



(a) (b)

Fig. 7. Eye movement data for the same task performed by multiple participants shown as a gaze plot (a) and in the STC (b)

Fig. 8 illustrates a few additional visual analytics functions of the STC. The moveable base map can be seen in Fig. 8a. Fig. 8b shows the option to freely rotate the cube to get views from the different viewpoints. Users can switch on or off part of path above or below the map (Fig. 8b). In Fig. 8c, we see that a user can define AOIs represented as boxes in the STC to focus on a section of the path. A temporal selection has been added as shown in Fig. 8d. In this case, one is focusing on spatio-temporal patterns in the defined AOI within a certain time interval. The figure focuses

on the fixations of participants in the defined AOI and in a certain time interval. With support of moveable base map feature, this AOI may be positioned directly on the context of stimulus.

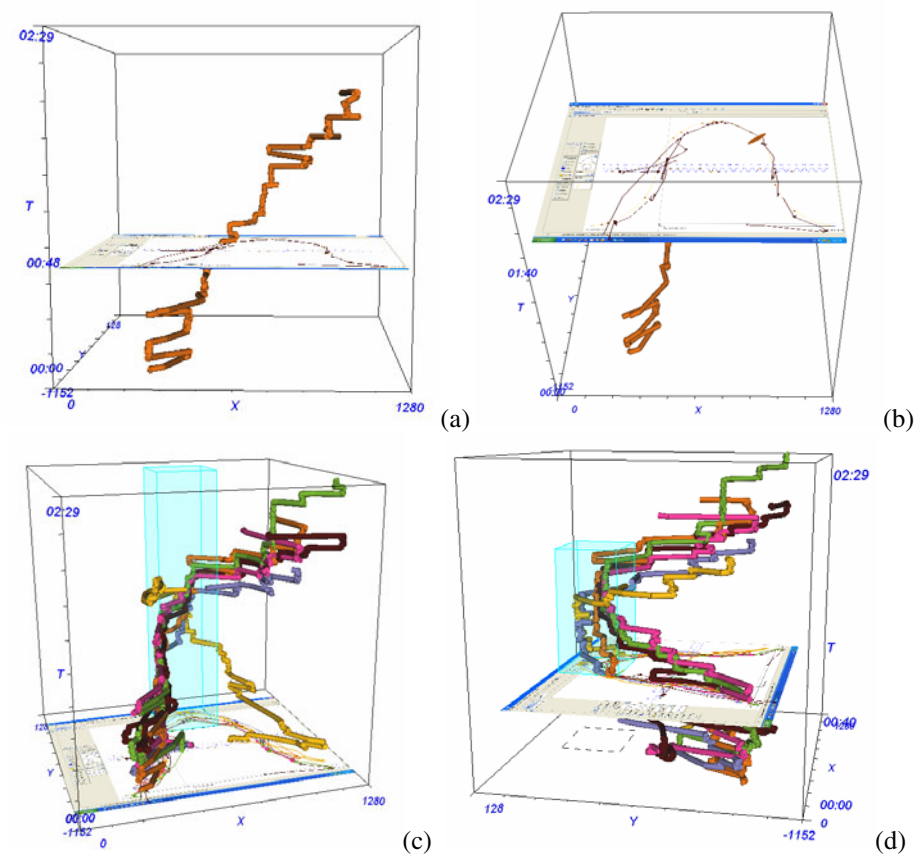


Fig. 8. A few additional visual analytics functions of the STC: (a) The moveable base map; (b) switching on or off the part of path above or below the map; (c) definition of the AOI (spatial selection) in STC; (d) temporal selection added to the previous spatial selection.

5 Discussion and Conclusions

Recording and analysis of eye movements offer interesting opportunities to support user-experience studies, including evaluation of geovisualization systems. Current eye movement recording and analysis hardware and methods have come a long way since the first studies in 1960s [52]. However, even with today's more comfortable procedures, analysis stages are still cumbersome. Qualitative analysis of data is typically

performed by visually inspecting gaze plots and density maps. Such representations are of course useful depending on the purpose [18], however, gaze plots typically suffer from massive overploting, and density maps offer only aggregate visualizations. Since the eye movement recordings essentially produce multivariate spatio-temporal data, geo-visual analytics methods that handle multivariate spatio-temporal data can be used to also analyze eye movement recordings. The STC is a 3D visualization method, which provides a combined view of time and space. While dynamic and interactive 3D visualizations may not be always easier to use for everyone, in particular novice users [18], patterns that can be discovered using the STC may offer new and/or a deeper understanding of eye movement data. These patterns will not be discovered as efficiently when spatial and temporal features are viewed separately, which makes the research on using the STC for eye movement analysis worthwhile. Exploration of eye movement data in the STC could also be useful for a quick overall understanding of the experiment, tasks and participant behavior. Many of the eye movement metrics integrated in usability studies (such as fixation and gaze durations, area of interest (AOI) analysis, and scan path comparisons) could be represented in modern STCs that allow multimedia integration, interactivity and dynamism. Furthermore, overlaps are avoided in the STC by extending the data onto the temporal dimension. At the same time, temporal order is visualized with the time dimension in the STC. Time-related questions regarding eye movements, such as *how long*, *how often*, *when* and *in what order* could be answered using these methods.

Developments in graphics processing, computer science and geovisualization domains provide even more opportunities for the use of the STC for eye movement analysis. Benefiting from these developments, several functions in recent STC software can help dealing with the difficulties in eye movement data analysis. For example, after an AOI is defined by a spatial zoom function, spatio-temporal behavior of a user's eye movement in this AOI could be explored further. A moveable base map along the time axis or a linked 2D view could offer more insight into the context of fixated regions. Dynamically changing the base map as the gaze plots are viewed in 3D space is potentially a great help with dynamic stimuli. Paths with annotations could provide useful information for user behavior analysis, such as videos, pictures and statistical graphs. Being able to change a view point in 3D space to explore complex data may provide additional insight into the complex multivariate data.

The use of the STC for visual analysis of eye movement data may complement statistical testing. To fully judge how useful these visualizations are, a good next step could be a usability study with experts working with eye tracking data. There are of course many other research questions that may follow up this study. For example, further visual inspection methods can be employed to study the spatio-temporal patterns found in eye movement data. Additionally, exploring the path annotations with qualitative analysis procedures could be taken into account. More importantly, to understand the thought processes better, not only *where* and *how long* but also *what* and *why* questions should be considered.

Acknowledgements. We would like to thank our three anonymous reviewers and Sara Fabrikant for their very helpful comments. Their constructive feedback greatly improved our paper.

References

1. Fabrikant, S.I., Rebich-Hespanaba, S., Andrienko, N., et al.: Novel Method to Measure Inference Affordance in Static Small-Multiple Map Dispalys Representating Dynamic Process. The Cartographic Journal 45(3), 201–215 (2008)
2. Çöltekin, A., Heil, B., Garlandini, S., et al.: Evaluating the Effectiveness of Interactive Map Interface Designs: A Case Study Integrating Usability Metrics with Eye-Movement Analysis Cartography and Geographic Information Science. 36(1), 5–17 (2009)
3. Brodersen, L., Anderson, H.H.K., Weber, S.: Applying Eye-Movement Tracking for the Study of Map Perception and Map Design. Kort & Matrikelstyrelsen, Denmark (2002)
4. Fuhrmann, S., Ahonen-Rainio, P., Edsall, R., Fabricant, S.I., Koua, E.L., Tobon, C.: Making useful and useable geovisualization:design and evaluation issues. Exploring Geovisualization (2004)
5. Haklay, M., Zafiri, A.: Usability engineering for GIS: Learning from a screenshot. Special Issue on Use and User Issues The Cartographic Journal 45(2), 87–97 (2008)
6. Goldberg, J.H., Kotval, X.P.: Computer interface evaluation using eye movements: methods and constructs. International Journal of Industrial Ergonomics 24, 631–645 (1999)
7. Cowen, L.: An Eye Movement Analysis of Web-Page Usability. Lancaster University, UK (2001)
8. Gitelman, D.R.: ILAB: A prog ram for postexperimental eye movement analysis. Behavior Research Methods, Instruments, & Computers 34(4), 605–612 (2002)
9. Ponsoda, V., Scott, D., Findlay, J.M.: A probability vector and transition matrix analysis of eye movements during visual search. Acta Psycholgica 88, 167–185 (1995)
10. Scinto, L., Barnette, B.D.: An algorithm for determining clusters, pairs and singletons in eye-movement scan-path records. Behavior Research Methods, Instruments, & Computers 18(1), 41–44 (1986)
11. Hägerstrand, T.: Spatial Process. University of Chicago, Chicago (1967)
12. Garlandini, S., Fabrikant, S.I.: Evaluating the Effectiveness and Efficiency of Visual Variables for Geographic Information Visualization. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 195–211. Springer, Heidelberg (2009)
13. Fabrikant, S.I., Rebich-Hespanha, S., Hegarty, M.: Cognitively inspired and perceptually salient graphic displays for efficient spatial inference making Annals of the Association of American Geographers 100(1), 13–29 (2010)
14. Jacob, R., Karn, K.: Eye tracking in humancomputer interaction and usability research: Ready to deliver the promises. In: Hyönä, J., Radach, R., Deubel, H. (eds.) The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research, pp. 573–605. Elsevier Ltd., Amsterdam (2003)
15. Henderson, J.M., Ferreira, F.: Scene Perception for Psycholinguists. In: Henderson, J.M., Ferreira, F. (eds.) The Integration of Language, Vision, and Action: Eye Movements and the Visual World. Psychology Press, New York (2004)
16. Rayner, K.: Eye Movements and Visual Cognition: Scene Perception and Reading. Springer, New York (1992)
17. Wade, N., Tatler, B.: The Moving Tablet of the Eye: The Origins of Modern Eye Movement Research. Oxford University Press, Oxford (2005)
18. Keehner, M., Hegarty, M., Cohen, C.A., Khooshabeh, P., Montello, D.R.: Spatial reasoning with external visualizations: What matters is what you see, not whether you interact. Cognitive Science 32, 1099–1132 (2008)

19. Goldberg, J.H., Schryver, J.C.: Eye-gaze determination of user intent at the computer interface. In: Findlay, J.M., Walker, R., Kentridge, R.W. (eds.) Eye Movement Research: Mechanisms, Processes and Applications, pp. 491–502. North-Holland Press, Amsterdam (1993)
20. Mackworth, N.H.: Stimulus density limits the useful field of view. In: Monty, R.A., Senders, J.W. (eds.) Eye Movements and Psychological Processes. Erlbaum, Hillsdale (1976)
21. Robinson, G.H.: Dynamics of the eye and head during movement between displays: A qualitative and quantitative guide for designers. Human Factors 21(3), 343–352 (1979)
22. Crowe, E.C., Narayanan, N.H.: Comparing interfaces based on what users watch and do. In: Proceedings Eye Tracking Research and Applications Symposium, pp. 29–36. Association for Computing Machinery, New York (2000)
23. Benel, D.C.R., Ottens, D., Horst, R.: Use of an eye tracking system in the usability laboratory. In: Human Factors Society 35th Annual Meeting. Human Factors and Ergonomics Society, Santa Monica (1991)
24. Jacob, R.J.K.: The use of eye movements in human-computer interaction techniques: what you look at is what you get. computer—human interaction 9(2), 152–169 (1991)
25. Ellis, S., Candrea, R., Misner, J., et al.: Windows to the soul? What eye movements tell us about software usability. In: Usability Professionals' Association Conference 1998, pp. 151–178 (1998)
26. Just, M.A., Carpenter, P.A.: Eye fixations and cognitive processes. Cognitive Psychology 8, 211–222 (1976)
27. Loftus, G.R., Mackworth, N.H.: Cognitive determinants of fixation location during picture viewing. Journal of Experimental Psychology: Human Perception and Performance 4(4), 565–572 (1978)
28. Steincke, T.R.: Eye Movement Studies In Cartography And Related Fields Cartographica. The International Journal for Geographic Information and Geovisualization 24(2), 40–73 (1987)
29. Belofsky, M.S., Lyon, D.R.: Modeling eye movement sequences using conceptual clustering techniques. Air Force Systems, Brooks Air Force Base: Air Force Human Resources Laboratory (1988)
30. Krose, B.J.A., Burbeck, C.A.: Spatial interactions in rapid pattern discrimination. Spatial Vision, 4211–4222 (1989)
31. Pillalamarri, R.S., Barnette, B.D., Birkmire, D., et al.: Cluster: a program for the identification of eye-fixation-cluster characteristics. Behavior Research Methods, Instruments, and Computers 25(1), 9–15 (1993)
32. Fabrikant, S.I., Hespanaha, S.R., Montello, D.R., et al.: A visual analytics approach to evaluate inference affordance from animated map displays. In: GIScience 2008 Pre-Conference workshop on Geospatial Visual Analytics, Park City, UT (2008)
33. Kraak, M.-J.: Cartography: Visualization of geospatial data, 2nd edn. (1996)
34. Hägerstrand, T.: How about People in Regional Science? Papers of the Regional Science Association 24, 7–21 (1970)
35. MacEachren, A.M.: How Maps Work: Representation, Visualization, and Design. Guilford press, New York (1995)
36. Andrienko, N., Andrienko, G.: Interactive maps for visual data exploration. International Journal Geographical Information Science 13, 355–374 (1999)
37. Hedley, N.R., Drew, C.H., Lee, A.: Hagerstrand Revisited: Interactive Space-Time Visualizations of Complex Spatial Data. Informatica: International Journal of Computing and Informatics (2), 155–168 (1999)

38. Kraak, M.-J.: Geovisualization illustrated. ISPRS Journal of Photogrammetry and Remote Sensing (5-6), 390–399 (2003)
39. Gatalsky, P., Andrienko, N., Andrienko, G.: Interactive Analysis of Event Data Using Space-Time Cube. In: Eighth International Conference on Information Visualisation (IV 2004), London, England, pp. 145–152 (2004)
40. Kraak, M.-J.: A visualization environment for the space-time-cube. Geo-Information Processing. Enschede (2004)
41. Kraak, M.-J.: Timelines, Temporal Resolution, Themporal Zoom and Time Geography. In: Proceedings 22nd International Cartographic Conference, A Coruna Spain (2005)
42. Kraak, M.J.: Geovisualization and time - new opportunities for the space-time cube. In: Dogde, M., Turner, M. (eds.) Geographic Visualization: Concepts, Tools and Applications, pp. 293–306. Wiley, New York (2008)
43. Johnson, I.: Mapping the fourth dimension: the TimeMap project. In: Dingwall, L., Exon, S., Gaffney, V., Laflin, S., van Leusen, M. (eds.) Archaeology in the Age of the Internet British Archaeological Reports International Series, vol. 21 (1999)
44. Forer, P., Huisman, O.: Computational agents and urban life spaces: a preliminary realisation of the time-geography of student lifestyles. In: Third International Conference on GeoComputation, Bristol (1998)
45. Kraak, M.J., Madzudzo, P.: Space Time Visualization for Epidemiological Research. In: Proceedings 23e International Cartographic Conference, Moscow (2007)
46. Kraak, M.-J., He, N.: Organizing the neo-geography collections with annotated space-time paths. In: The 24th International Cartographic Conference, Chile (2009)
47. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages, pp. 336–343. IEEE Computer Society Press, Piscataway (1996)
48. Peuquet, D.J.: Representations of Space and Time. Guilford, New York (2002)
49. Li, X., Kraak, M.-J.: he Time Wave. A New Method of Visual Exploration of Geo-data in Time–space. The Cartographic Journal 45(3), 1–9 (2008)
50. GIVA's Eye Movement Lab, `http://www.geo.uzh.ch/en/units/gisience-giva/services/eye-movement-lab/`
51. uDig, `http://udig.refractions.net`
52. Yarbus, A.L.: Eye Movements and Vision. Plenum Press, New York (1967)

5D Data Modelling: Full Integration of 2D/3D Space, Time and Scale Dimensions

Peter van Oosterom[1] and Jantien Stoter[1,2]

[1] OTB, GISt, Techncial University of Delft, The Netherlands
[2] Kadaster, Apeldoorn, The Netherlands
{p.j.m.vanoosterom,j.e.stoter}@tudelft.nl

Abstract. This paper proposes an approach for data modelling in five dimensions. Apart from three dimensions for geometrical representation and a fourth dimension for time, we identify scale as fifth dimensional characteristic. Considering scale as an extra dimension of geographic information, fully integrated with the other dimensions, is new. Through a formal definition of geographic data in a conceptual 5D continuum, the data can be handled by one integrated approach assuring consistency across scale and time dimensions. Because the approach is new and challenging, we choose to step-wise studying several combinations of the five dimensions, ultimately resulting in the optimal 5D model. We also propose to apply mathematical theories on multidimensional modelling to well established principles of multidimensional modelling in the geo-information domain. The result is a conceptual full partition of the 3Dspace+time+scale space (i.e. no overlaps, no gaps) realised in a 5D data model implemented in a Database Management System.

Keywords: Multidimensional data modelling, spatial DBMSs, spatial data types, spatio-temporal data models, multi-scale data models, 3D data models.

1 Introduction

The role of geographic information in our society has changed tremendously the past decades. From being collected and used ad hoc for maps and other specific applications, it has now become part of the Geo Information Infrastructure (GII). Ultimately GIIs will serve complete information flows in the web from observing/monitoring via processing/analyzing/planning to communicating and controlling actions. After many isolated initiatives, our society is now heading towards a sustainable GII in which geo-data is shared and re-used by many and highly different users and applications through machine-based linking of large amounts of distributed geographic data and information.

Formal definitions of geo-information are required to enable understanding and satisfying the requests of people and (increasingly) machines to use appropriate geo-information available within the web. This paper focuses on formalising the data models and structures that capture geo-information. Data structures for geo-information bring specific challenges as traditional DataBase Management System (DBMS) implementations for non-dimensional information are not capable of

S.I. Fabrikant et al. (Eds.): GIScience 2010, LNCS 6292, pp. 310–324, 2010.

handling the different dimensions of geo-data sufficiently. In our approach we distinguish five dimensional concepts of geo-data.

Apart from their location and 0D to 3D geometrical and topological characteristics, geo-data has further temporal (when was a moving object at a specific location; when was an object valid in the database?) and scale components that were often implicitly taken into account when the data was collected. These different dimensional aspects highly correspond, e.g. a (possibly geometric) change may be only relevant for the highest scale of an object or understanding the route directions for a long car trip requires overview, but at specific locations (e.g. to rest or to stay overnight) consistent information at a higher scale, with also temporal information (i.e. weather conditions at a certain moment) may be needed. Although (multi-)scale is a well-known concept in the geo-information technology domain, regarding it as an extra dimension of geo-data, integrated with the other dimensions, is new.

Despite the interdependencies, until now different dimensions of geo-data have been studied in separate initiatives, with sometimes limited support for the other dimensions. Although these past studies have gained important knowledge on how to handle the individual dimensions 2D/3D space, time and scale, no modelling approach exists that truly integrates all dimensional concepts of geo-data at a fundamental level.

This is our motivation to start a new research on a conceptual full partition of 3Dspace+time+scale (i.e. no overlaps, no gaps) realised in a true 5D generic model. This paper elaborates on the research methodology that we propose to accomplish the true 5D model. In contrast to a separate handling of spatial, temporal and scale dimensions, a true 5D approach provides a sustainable and solid foundation for the GII for three main reasons:

- The deep integration of all dimensional concepts accomplishes a highly formal definition of geo-data (with 5D data types and 5D topological primitives) as the relationships between space, time and scale aspects of geo-data are fully addressed and no special cases need to be treated in another way anymore. Every case will be a specialisation of the model.
- The model enforces consistency crossing dimensional borders which improves the quality of geo-data.
- Optimal efficient 5D searching and maintenance can only be realised if a 5D data type and index/clustering is used, otherwise the DBMS query plan has to select first on space, then on time and then on scale (or in another order). An example of a 5D search that appears in space, time, and scale context is the integration of a database with physical plans at different moments in history and a database with historical information on buildings to check which buildings (extensions) conflict with which status of the physical plan. Another example of a 5D search is comparing the cadastral database that registers the legal status of networks based on the physical extent of the network at the moment of registration with the physical registration maintained by the network company in which changes as extensions, deletion and movements of parts of the network are recorded as well.

In our approach the multidimensional integration is studied at two levels. At first a conceptual 5D data model will be designed on which all other geo-data models can be founded. Secondly, as the foundation of the GII consists of geo-DBMSs maintaining

geographic information, the methodology proposes to study how DBMS functionality can be extended up to 5D as implementation of the conceptual model.

We do recognise the high ambitions to realise a true 5D data model. However our aim is to lay down a foundation for multidimensional data modelling by defining a theory validated through prototype implementations, which can be further developed in the future. In addition, in a step-wise approach we will apply mathematical theories on multidimensional modelling to established principles in 2D/3D, time and multi-scale modelling. By studying several combinations of the different dimensions, we will accomplish the optimal method for including multidimensional concepts and notions in geo-data modelling.

To explain our proposed methodology in which we combine multidimensional principles established in both the geo-domain and mathematical theories, we first elaborate on the multidimensional modelling concepts and principles that are studied in the geo-information domain (section 2), while in section 3 we will explain the potential mathematical theories on multidimensional modelling that we will explore. Section 4 explains our proposed methodology in more detail and the paper ends with discussion in section 5.

2 Principles of Multidimensional Modelling of Geo-information

This section firstly describes in Section 2.1 how earlier work studied the various dimensional aspects of geo-information. Sections 2.2 to 2.4 describe the principles established in 2D/3D, spatio-temporal and multi-scale modelling separately, which form the point of departure for the true 5D data model in our research.

2.1 Dimensional Aspects of Geo-information in Previous Work

The high correspondence between 2D/3D spatial, time and scale characteristics of geo-information has been recognised by other researchers. A well-known example is [1] who studied the question 'How long is the coast of Britain'. Based on empirical evidence he found out that the measured length of a coastline depends on the scale of measurement: the smaller the increment of measurement, the longer the measured length becomes. This is because smaller increments allow a more curvilinear route. The coastline example can be realistically extended into 5D by including 3D space and time (besides the scale): what is the volume of above sea-level a 100m wide 'strip'(inland) and how is this evolving over the last 50 years?

Other researches that studied the varying perception and meaning of concepts dependent on the scale of observation are [2, 3, 4, 5, 6, 7]. We propose a data model and structure that makes these dependencies explicit.. This will make it possible to fully support 5D applications, for example efficient and consistent monitoring of coastline change at different scales.

Apart from initiatives that focused on the scale dimension of geo-information, several previous researches have studied multidimensional modelling of geo-information.

A first related topic is nD storage and mining [8, 9] which aims at aggregating information on multiple thematic (non-spatial) attributes to perform efficient database querying subsequently. For example 5D data would be the result of combining object-id,

weight, colour, price, and date attributes. Although this 5D data focuses on thematic attributes in data mining and not on dimensional concepts in geo-data modelling, the similarity is that it also considers multiple aspects of data in an integrated manner.

Also the domain of nD modelling is related [10, 11]. nD modelling extends BIMs (Building Information Models) with additional thematic characteristics to serve each stage of the lifecycle of a building facility through one common information model. However nD modelling focuses mainly on integrating multiple thematic, and not multidimensional, concepts. As the multidimensional data model that we propose offers a framework to structure any geo-information, the thematic approach of nD modelling can be well served by progresses in multidimensional data modelling, for example 4D BIMs that include the time dimension.

2.2 2D/3D Modelling

In 2D/3D spatial modelling we can observe the following developments.

The Open Geospatial Consortium (OGC) and ISO establish standards for handling spatial data, which has resulted in Simple Features Specifications for Structured Query Language (SQL) in 2003 [12, 13, 14]. These specifications define how to support 0D, 1D and 2D spatial objects (that can be defined in 2D and 3D space) in object-relational DBMS environments. The specifications also define operations to detect eight topological relationships defined in the 9-intersection framework of [15], i.e. equals, disjoint, intersects, touches, crosses, within, contains and overlaps. Mainstream DBMSs have implemented these specifications, resulting in a shift from ad hoc use of geo-data to interoperable geo-data as part of generic information flows.

Currently no standards exist for implementing 3D geometry and topology in DBMSs. However 3D data structures achieved in research have shown that geometry and topology of 3D objects can be structured in several ways: polyhedral [16, 17, 18] regular polytopes [19] and TEN (Tetrahedral Network) [20]. More research is required to see what applications can be served best by what kind of 3D data structure, how to define standards for more advanced 3D geometry, 3D topological primitives and 3D topological relationships and how to enforce the validity of such data, see also [21, 22, 23, 24].

Many initiatives have studied modelling the 3D concepts of geo-data in information models driven by applications such as facility management, urban planning, 3D cadastre, noise modelling, flooding, disaster and crisis management. To unify these initiatives, OGC and ISO TC 211 established a standard for exchanging 3D city information in 2008 [25, 26, 27, 28]. This information model, called CityGML and based on the Geography Markup Language (GML), provides a common definition and understanding of basic entities, attributes and relationships of 3D city objects. GML provides classes for 0D to 3D geometric primitives, 1D-3D composite geometries (e.g. *CompositeSurface*), and 0D-3D geometry aggregates (e.g. *MultiSurface* or *MultiSolid*). The geometry in CityGML follows the ISO 19107. Generally volumetric objects are possible, but the validity of closed volumes cannot be enforced in CityGML. In addition 3D topological structures are not standardised in CityGML. Currently CityGML only defines relationships between face-sharing features via a reference to a common outer shell polygon and between feature classes and the terrain

through the terrain intersection curve [28]. Also the vector (and specifically simple feature) representations may not be suitable to represent every type of 3D object [29].

The time and scale dimensions are handled in CityGML, however not in an integrated manner. The time dimension is separately handled by adding attributes to geometrical objects, i.e. *creationDate* and *terminationDate*. Scale is handled via the Level-of-Detail (LoD) concept. CityGML models multiple geometrical representations of the same real world with increasing accuracy and structural complexity (LoD0 to LoD4). However the different LoDs are poorly connected and therefore consistency between different LoDs cannot be assured. In addition different LoDs are maintained in different representations, i.e. on the fly derivation of lower LoDs from a higher LoD is not supported and higher LoDs cannot consist of parts from a lower LoD [30]. A final shortcoming of LoDs in CityGML is that the indoor-level (LoD4) is not well represented. For example what is the inside of building: a hole (inner polygon) in the building object or another world that should have its own LoDs [30, 31]?

2.3 Spatio-temporal Modelling

Temporal aspects of geo-data is fundamental for recording or monitoring changes, for describing processes, and for documenting future plans. For example monitoring the status change of a set of related features (Figure 1, left) or monitoring changes of moving objects (Figure 1, right).

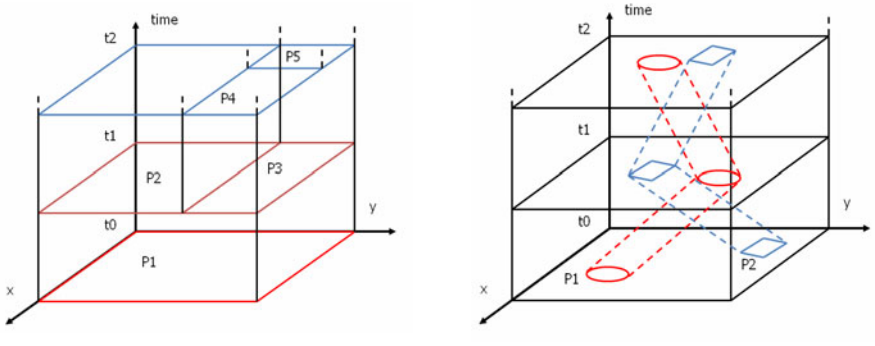


Fig. 1. Time as third dimension: division of parcels (left) and moving objects (right) [32]

Many Spatio-Temporal (ST) data models have been designed to model changes of geo-information [33, 34, 35]. The semantics of the time dimension included in these models vary from model to model and generally address the following items:

Temporal granularity specifies to which units of data one temporal attribute is added, e.g. whole dataset, object class, object instance or attribute.
Temporal operations for spatio-temporal analyses.
Modelling foundation for time describes which type of changes can occur to the value of a thematic or geographic characteristic, i.e. discrete changes or more continuous/gradual change.
System (or transaction) time indicates the time an event is recorded in the database.

Valid (or real-world) time describes the time that an event happened in the real world.
Lifespan identifies the history track of real world objects. Some events last only one short moment, e.g. an explosion or a traffic accident, which are like point objects. Other situations last for a longer period of time, e.g. the fact that a building has a particular owner, which are like linear objects representing a time interval.
Representation of time can differ from maintaining the duration of the status of an object (i.e. period) to recording events (i.e. start- and end-moment) that imply status change.

For ten well-known ST models, Table 1 shows how they represent the time dimension [36].

Table 1. Representation of time in ten well-known ST models, after [36]

Spatio-Temporal Data Models	***Representation of Time***	
	Models	***Time as***
Snapshot model	Layers- Snapshots	Attribute of location
Space-Time Composite	Polygon history	Integral part of spatial entities
Simple Time stamping	Object's Creation – Cessation	Attribute of the object
Event Oriented	Events, change	Attribute of an event
Semantics, space and time separately in 3 Domain	Temporal versions	Independent object
History Graph	Events, processes	Attribute of objects, events
ST Entity Relationship (STER)	Entity change	Attribute of entity, relationship
Object Relationship Model	ST phenomena	Attribute of Object
ST Object-Oriented	Object Change	Attribute of object
ST UML	TimeUnit	Via the Specification box
Moving Object Data Models	Functions	Integral part of spatial entities

The deep integration of time with space and scale concepts will fully handle changes upon position, attributes and/or extent of the objects in the unified space-time-scale continuum. Some aspects require specific attention for the time dimension in this deep integration. At first all possible changes of geo-information at varying scales should be well represented, i.e. change in geometry OR topology OR attributes, or no change at all. In addition the integrated representation of time should not only support changes at discrete moments, as currently supported by most of the ST models via timestamps and versioning, but also continuous temporal changes to describe the movement or change of objects independently from their object identification. Also the integrated space-time-scale approach requires specific attention for topological relationships between (continuously) evolving geographic objects. It should be noted that temporal modelling itself also has a scale dimension, i.e. at what level of detail the time dimension (temporal resolution) is represented. This relates to temporal granularity as explained above. In our research we will pay specific attention to allow modelling multi-scale time dimension and the interplay with the multi-scale geometric dimensions.

More researchers have identified the need for a generic spatio-temporal data model. A theory on a unified spatial-temporal data model was proposed in [37] and a

generic spatio-temporal data type in a relational DBMS was suggested in [38] and [39]. We will use these studies to model the time dimension with the other dimensional concepts of geo-data, also incorporating the syntax for (fuzzy) time dimension as specified by ISO for Geographic Data Files (GDF) in the transport domain [40].

From the above it becomes clear that several temporal aspects can be relevant and therefore they may question our approach to treat time as one dimension. Specifically, the bi-temporal model including both the real-world and the system time, is a good example showing that time is more than just a single dimension. However, in this paper we limit ourselves to treat only one temporal dimension in the 5D model/structure (without overlaps and gaps). Most likely this will be the system time and other temporal attributes will be treated as normal, non-integrated, attributes (for the time being). This is motivated by the assumption that for system time it is more feasible to avoid the overlaps and gaps.

2.4 Multi-scale Modelling

Modelling different scales of geo-data is related to the "coarse-to-fine" hierarchical structure of how we perceive, model and understand our environment. In some applications less detailed, but simpler data works better, especially when there is a need for an overview. In other cases very detailed data is required.

Two basic options exists for maintaining data sets of the same real world at different scales. First option is to separately maintain different databases at predefined scale-steps. This option is practiced by many National Mapping Agencies that produce maps at different scales. Second option is to maintain only the most detailed data and to automatically generalise small scale data from it on the fly, eventually supported by pre-storing the results of costly geometric computations in multi-representation (as in the first option).

To provide and reuse multi-scale data within the GII, consistency between data at different scales is fundamental, i.e. the availability of data at different scales free from contradictions enabling smoothly zooming in and out. This is supported by multi-representation data models that formally define different scale states of the data.

Many researchers have studied multi-representation data models since it was introduced in [41, 42]. Examples are MRMS [43], MADS [44], Perceptory [45], modelling multiple geometries [46], modelling scale transitions between pairs of objects [47] and modelling links between instances [48]. While these previous initiatives mainly aimed at controlling the redundancy of multi-representations and multi-scale data, our study will specifically aim at reducing redundancy to improve efficiency and to better assure consistency between different scales.

Therefore we will first extend previous initiatives on multi-scale data modelling with an explicit notion of how geo-information changes at scale transitions. With this, we will build on the semantic rich information model presented in [49] that integrates the data states at different scales and at the same time formalises semantics on scale transitions. In addition we will build on the tGAP data structure [50, 51, 52], where the data structures can be queried based on the importance value of the objects, as shown in Figure 2.

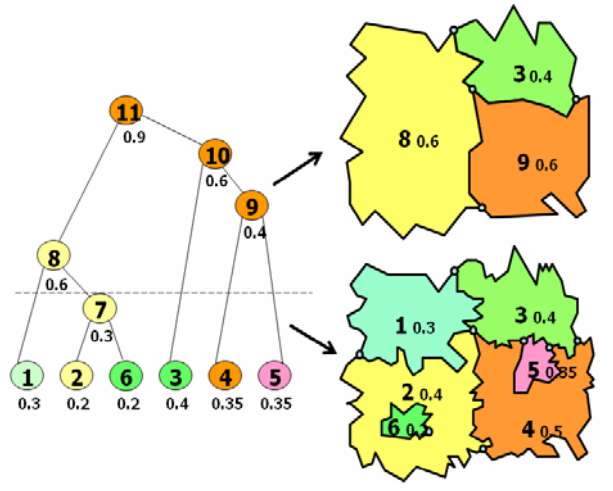


Fig. 2. Illustration of the working of the current tGAP structures

The tGAP data structure enables objects to be stored once and to be displayed at any arbitrary scale, supporting smooth zooming and progressive transfer. The idea of integrating 2Dspace and scale in the tGAP structure was first presented in [53]. A step-wise process will show how the tGAP structure can be extended to the time dimension to process time changes efficiently and to include the complete history in the same vario-scale structure. This will enable to query objects at any arbitrary scale and moment in time.

To embed the 3D scale concept in the tGAP data structure, our research will enrich the 3D Level-of-Detail (LoD) concept as studied in computer graphics with semantics on geo-information. In computer graphics LoD involves decreasing the complexity of a 3D object representation as it moves away from the viewer or according other metrics such as object importance or position.

3 Mathematical Theories on Multidimensional Modelling

To realise a 5D geo-data modelling approach by which the treatment of (up to 3D) space, time and scale is optimally integrated, we will study existing mathematical theories on multidimensional descriptions and apply them to the well defined frameworks for 3D, time and scale modelling in the geo-information domain. Examples of established mathematical theories on multidimensional modelling are:

- Topological polyhedra where multidimensional objects are built from their lower primitives, i.e. a 3D volume object consists of 2D faces that consist of 1D edges that consist of 0D nodes [54, 55].
- Regular polytopes, which is based on a division of space by hyperplanes, e.g. a 3D volume object is described by 2D planes [56].

- Simplicial Homology based n-simplexes, which are the building blocks for the Triangular Irregular Network (TIN in 2D) and Tetrahedral Network (TEN in 3D) and their higher dimensional equivalents [57].

The first theory is advantageous for multidimensional geo-data modelling because it aligns to the boundary representation of 3D volume objects of OGC [58]. However since it lacks of a well defined fundament, validity of objects has to be fully handled by additional functionalities. The advantage of the second theory is that the formalisation of multidimensional concepts is straightforward because the primitives that build an object are described with equations of the hyperplanes (and are valid in any dimension). Finally, the third theory is advantageous for geo-data modelling because of the n-simplex (e.g. triangle, tetrahedron) based approach. Triangles contain specific characteristics, such as convexity, that make it easy to enforce validity of objects that consist of the lower dimensional primitives (and again this theory is valid in any dimension).

Potentials of the three theories for handling multidimensional geo-data are shown by a) [16, 17, 18], b) [19], and c) [20] who implemented the respective three theories into a 3D data structure.

To explain how the simplicial homology theory may be applied to multidimensional geo-data modelling, we will now describe how it was applied in [20] resulting in a network of simplicial complexes forming a partition of space, i.e. a TEN in 3D (see Figure 3). The boundary of a n-simplex Sn is defined as the sum of n+1 simplexes of dimension n-1; e.g. the tetrahedron S3 (n=3) has 3+1=4 boundaries, which are of dimension 3-1=2 (triangles).

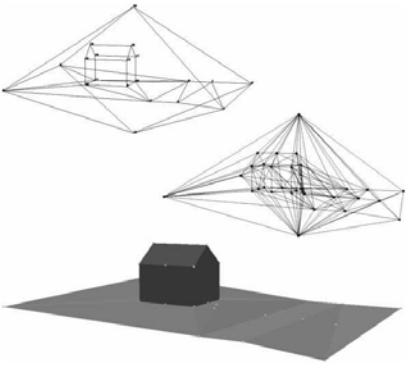


Fig. 3. Simple scene and the TEN

However, the theory is valid for any dimension, so S4 has 5 boundaries of dimension 3, that is 5 tetrahedrons. How would these fit in a network of simplicial complexes in 4D and how could this be used to present the tight integration of 3Dspace and time (or scale)?

Our research will further study how well the different mathematical approaches are fit for 5D data modelling and also whether other mathematical approaches might better fit.

4 Research Methodology for 5D Data Modelling

Because of the unexplored domain of deeply integrated 5D information modelling, much knowledge need to be gained on the optimal 5D approach. To do so, in our methodology we propose to first apply mathematical theories on multidimensional modelling to principles established in 2D/3D, spatio-temporal and multi-scale data models and to gradually extend the results with extra dimensions in three iterations (A, B and C), see Figure 4. This will lead to three alternative 3D models in the first iteration (3Dspace, 2Dspace+time, 2Dspace+scale) and three alternative 4D models in

the second iteration (3Dspace+time, 3Dspace+scale, 2Dspace+time+scale), finally leading to the best 5D data model in which lower dimensional objects are supported as well. The intermediate trajectory is important to optimally prepare the separate approaches for an integrated 5D data modelling approach and to gain fundamental knowledge on how to best address the different dimensions in the integration, both at conceptual model level and on database technology level.

The steps applied in every iteration are: 1. Conceptual modelling, 2. Implementation (with test data), and 3. Testing and validation (with real world scenarios).

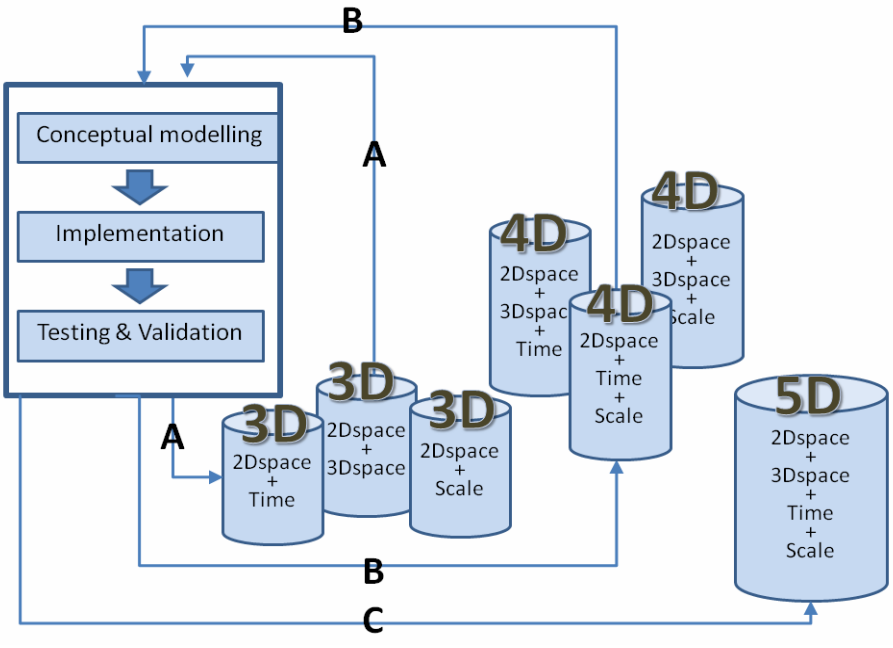


Fig. 4. Workflow of research methodology

Since *iteration A* starts from established principles, the resulting 3D models should be well feasible and in reach, also because they can be built on the partial models that are already operational, e.g. 3D in commercial systems (Bentley Systems, Oracle, ESRI), spatio-temporal databases and vario-scale data structures. Therefore the main aim of this iteration is gaining insight into how integrated spatial and time (or scale) dimensions behave when applying the multidimensional mathematical theories (simplicial homology, regular polytopes).

The combined dimensions of *iteration B* belong to hardly explored types of models and implementations, but we expect them to be feasible (in part). In addition they will provide insight in even more complicated integrated 5D modelling, which is still required as these 4D models focus on a selection of multidimensional concepts only. Again these explorations will provide more insight in the behaviours of integrated dimensions.

Iteration C will use the knowledge from iterations A and B to generate concepts for the 5D data model deeply integrating space, time and scale concepts implemented with a 5D data type, 5D topological structures and primitives as well as 5D clustering and indexing.

For validating the (intermediate) research results we will establish application tests with large datasets containing 2Dspace and 3Dspace geo-information at several scales also containing time information. For these tests we can make use of the spatio-temporal cadastral database and of large, mid- and small-scale topographic datasets of the Netherlands' Kadaster. In addition various large 3D datasets are available, such as the Actual Height Model of The Netherlands, 3D datasets of municipalities (Amsterdam, Rotterdam and Tilburg) and a 3D detailed topographic dataset of Rijkswaterstaat.

5 Discussion

This paper proposes a research methodology for 5D data modelling that fully integrates 2D/3D space, time and scale aspects of geo/information. The methodology combines established principles on 2D/3D space, time and scale modelling in the geo-information domain with mathematical theories on multidimensional modelling. Although 3D, time and scale aspects have been studied in separate research domains, studying the deep integration of time and scale concepts in the traditional 2D/3D models to replace their separate time and scale treatment with a full partition of 3Dspace+time+scale is new. Unique is also the approach to combine both fundamental theories with information technology (i.e. DBMS implementation) in multidimensional data modelling. The approach will result in a new theory and method for geo-data modelling as well as technologies realising a multidimensional partitioning.

Integrating multidimensional concepts of geo-data enables shared geometry and embedded topological, temporal and scale structures through a full partition of 3Dspace+time+scale. Instances can be identified as separate features but they will not have independent 2D/3D, scale and temporal attributes. Instead they will refer to primitives in the full multidimensional continuum. The resulting model will contain a highly formal definition of the dimensional concepts of geo-data allowing optimal flexibility to define specific semantics for each feature type and each dimension separately.

As the research approach extends currently available single-dimensional models in a step-wise approach, the intermediate models that integrate multiple but not all dimensional concepts are already in reach for use in practice and commercial implementations within the next few years, i.e. several 3D models after one year (3Dspace, 2Dspace+time, 2Dspace+scale) and several 4D models after three years (3Dspace+time, 3Dspace+scale, 2D+time+scale), all based on a solid mathematical theory.

Several stakeholders will benefit from this research. A first group of stakeholders that will benefit are providers of geo-information for which the multidimensional data types provide important advantages with respect to efficiency and consistency compared to the current separate treatment of space, time and scale. A common characteristic of these providers is that they are responsible for maintaining and providing large amounts of geo-information at different scales for which it is increasingly important to keep history track record. An integrated approach for multidimensional concepts of geo-data enables these organisations to be optimally prepared for provision of geo-information in the Semantic Web in the future. A second group of stakeholders that

will benefit from the intermediate and final results are vendors of geo-ICT systems that can implement the multidimensional data types as realised in prototypes. A final group of stakeholders, and perhaps in the long term the most important group, that will benefit from this research are end-users of geo-information who will be served by improved and new 5D aware applications and services.

In the long term results on 5D data modelling in the geo-information technology domain are important for standards on geo-information which are established and developed by ISO TC 211 and the Open Geospatial Consortium.

Finally it should be noted (and it is quite well-known) that depending on the application, different types of objects may be more (or less) relevant than others. This then results in a different generalisation/scale structure. One could image having different 5D representations for different applications, all starting from the same base data. The existence and relative importance of the classes in the various applications could also be considered in a more integrated manner as the sixth dimension: the semantic-dimension. For, the time being this is considered out of our research scope (and single application profile is assumed).

References

1. Mandelbrot, B.: How Long Is the Coast of Britain? In: Statistical Self-Similarity and Fractional Dimension. Science. New Series, vol. 156(3775), pp. 636–638 (1967)
2. Fisher, P., Wood, J., Cheng: Where is Helvellyn? Fuzziness of multi-scale landscape morphometry. Transactions of the Institute of British Geographers 29, 106–128 (2004)
3. Levin, S.A.: The problem of pattern and scale in ecology. Ecology 73, 1943–1967 (1992)
4. Fisher, P.F., Wood, J.: What is a mountain? Or the Englishman who went up a Boolean geographical concept and realised it was fuzzy. Geography 8(3), 247–256 (1998)
5. Tate, N., Wood, J.: Fractals and scale dependencies in topography. In: Tate, N., Atkinson, P. (eds.) Modelling scale in geographical information science, pp. 35–51. Wiley, Chichester (2001)
6. Wood, J.: Scale-basedc haracterisationo of digital elevation models. In: Parker, D. (ed.) Innovations in GIS, vol. 3. Taylor & Francis, London (1996)
7. Wood, J.: Visualizing the structure and scale dependency of landscapes. In: Fisher, P., Unwin, D. (eds.) Virtual reality in geography, pp. 163–174. Taylor & Francis, London (2002)
8. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. In: Data Mining and Knowledge Discovery, vol. 1, pp. 29–53. Kluwer Academic Publishers, Dordrecht (1997)
9. Casali, A., Cicchetti, R., Lakhal, L.: The 3rd SIAM International Conference on Data Mining. Cube Lattices: a Framework for Multidimensional Data Mining, pp. 3004–3008 (2003)
10. Hamilton, A., Wang, H., Tanyer, A.M., Arayici, Y., Zhang, X., Song, Y.: From 3D to nD modelling. ITcon 10, 55–67 (2005)
11. Aouad, G., Lee, A., Wu, S.: Special issue on 'From 3D to nD modelling. Journal of Information Technology in Construction (2005)
12. OGC, The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema), Version 5. Technical Report OpenGIS Project Document. Wayland, Mass., VS, Open Geospatial Consortium. N 01-101 (2001)

13. ISO, ISO/TC 211, ISO International standard 19107:2003, Geographic information - Spatial schema (2003)
14. Herring, J.: Implementation Specification for Geographic information - Simple feature access- Part 2: SQL option. OGC 06-104r3 (2006)
15. Egenhofer., M.J.: Spatial SQL: A Query and Presentation Language. Transactions on Knowledge and Data Engineering 6, 86–95 (1994)
16. Arens, C.A., Stoter, J.E.: Modelling 3D spatial objects in a geo-DBMS using a 3D primitive. Computers&Geosciences, 165–177 (2005)
17. Brisson, E.: Representing geometric structures in d dimensions: Topology and order. In: Proceedings 5th Annual Symposium on Computational Geometry, pp. 218–227. ACM Press, New York (1989)
18. Pigot, S.: A topological model for a 3D spatial information system. In: Proceedings 5th International Symposium on Spatial Data Handling, pp. 344–359 (1992)
19. Thompson, R.J.: Towards a Rigorous Logic for Spatial Data Representation. PhD thesis. Delft University of Technology. Netherlands Geodetic Commission, p. 333 (2007)
20. Penninga, F.: 3D Topography A Simplicial Complex-based Solution in a Spatial DBMS, PhD thesis. TU Delft. The Netherlands, p. 204 (2008)
21. Raper, J., Livingstone, D.: Development of a geomorphological spatial model using object-oriented design. International Journal of Geographical Information Science 9, 359–383 (1995)
22. Raper, J.: Multidimensional geographic information science. Taylor&Francis, London (2000)
23. Ledoux, H., Gold, C.M.: Simultaneous storage of primal and dual three-dimensional subdivisions. Computers, Environment and Urban Systems 31, 393–408 (2007)
24. Kazar, B.M., Kothuri, R., van Oosterom, P., Ravada, S.: On Valid and Invalid Three-Dimensional Geometries. In: Advances in 3D Geoinformation Systems, ch. 2, pp. 19–46. Springer, Heidelberg (2008)
25. Gröger, G., Kolbe, T., Nagel, C., Czerwinski, A.: Open Geospatial Consortium City Geographic Markup Language. In: CityGML Encoding Standard document version 1.0.0, OGC 08-007r1 (2008)
26. Gröger, G., Kolbe, T., Czerwinski, A.: OpenGIS CityGML Implementation-Specification. OGC 06-057 (2006)
27. OGC, CityGML specification document version 0.4.0, approved Best Practice Paper, OGC 07-062 (2009), `http://www.opengeospatial.org/standards/bp`
28. Kolbe, T.H.: Representing and exchanging 3D city models with CityGML. In: Lee, J., Zlatanova, S. (eds.) Proceedings of 3rd International Workshop on 3D Geo-information. Lecture notes Geoinformation&Cartography. Springer, Heidelberg (2008)
29. Tegtmeier, W., Zlatanova, S., van Oosterom, P.J.M., Hack, H.R.G.K.: Information management in civil engineering infrastructural development: with focus on geological and geotechnical information. In: Zhang, T.H., Kolbe, S.Z. (eds.) Proceedings of the ISPRS workshop, vol. XXXVIII-3-4/C3 Commission III/4, IV/8 and IV/5, Berlin (2009)
30. Isikdag, U.: Towards the implementation of building information models in geospatial context, PhD Thesis. 3D geo-information sciences. UK: The research institute for Built and Human Environment, University of Salford (2006)
31. Emgård, K.L., Zlatanova, S.: Design of an integrated 3D information model. In: Rumor, F., Coors, Z. (eds.) UDMS annual, pp. 143–156. Taylor & Francis Urban and regional data management, London (2008)
32. Oosterom, P.J.M., Ploeger, H., Stoter, J., Thompson, R., Lemmen, C.: Aspects of a 4D Cadastre: A First Exploration. In: FIG congress, Munich, Germany (2006)

33. Hornsby, K., Egenhofer, M.J.: Identity-based change: a foundation for spatio-temporal knowledge representation. International Journal of Geographical Information Science 14, 207–224 (2000)
34. Peuquet, D.J.: Representations of Space and Time, p. 394. Guilford, New York (2002)
35. Raper, J.F., Livingstone, D.E.: Let's get real: spatio-temporal identity and geographic entities. Transactions of the Institute of British Geographers 26, 237–242 (2001)
36. Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y.: Literature Review of Spatio-Temporal Database Models. The Knowledge Engineering Review journal 19, 235–274 (2004)
37. Worboys, M.F.: A unified model for spatial and temporal information: Spatial data: applications, concepts, techniques. Computer journal 37, 26–34 (1994)
38. Jin, P., Yue, L., Gong, Y.: Research on a Unified Spatiotemporal Data Model. In: International Symposium on Spatial-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion. ISPRS Press, China (2005)
39. Oosterom, Van, P.J.M., Maessen, B., Quak, C.W.: Generic query tool for spatio-temporal data. International Journal of Geographical Information Science 16, 713–748 (2002)
40. ISO, ISO 14825:2004 Intelligent transport systems – Geographic Data Files (GDF) – Overall data specification. ISO Technical Commission on Intelligent transport systems, p. 590 (2004)
41. NCGIA, National Center for Geographic Information and Analysis, The research plan of the National Center for Geographic Information and Analysis. International Journal Geographical Information Systems 3, 117–136 (1989)
42. Buttenfield, B.P., Delotto, J.S.: Multiple representations. Scientific Report for the Specialist Meeting. National Center for Geographic Information and Analysis (NCGIA), p. 87. Technical paper 89–3 (1989)
43. Friis-Christensen, C.S., Jensen, A.: Object-relational management of multiply represented geographic entities. In: Proceedings of the 15th International Conference on Scientific and Statistical Database Management., Cambridge, MA, USA, July 9–11 (2003)
44. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual modelling for traditional and spatio-temporal applications. In: The MADS approach. Springer, Heidelberg (2006)
45. Bédard, Y., Larrivée, S., Proulx, M.-J., Nadeau, M.: Modelling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on Perceptory. In: Wang, S., Tanaka, K., Zhou, S., Ling, T.-W., Guan, J., Yang, D.-q., Grandi, F., Mangina, E.E., Song, I.-Y., Mayr, H.C. (eds.) ER Workshops 2004. LNCS, vol. 3289, pp. 17–30. Springer, Heidelberg (2004)
46. Jones, C.B., Kidner, D.B., Luo, L.Q., Bundy, G.L., Ware, J.M.: Database design for a multi-scale spatial information system. International Journal Geographic Information Science 10, 901–920 (1996)
47. Devogele, T., Trevisan, J., Raynal, L.: Building a multi-scale database with scale transition relationships. In: International Symposium on Spatial Data Handling, pp. 337–351 (1996)
48. Kilpelainen, T.: Multiple representation and generalisation of geo-databases for topographic maps. PhD thesis. Finnish Geodetic Institute (1997)
49. Stoter, J.E., van Oosterom, P.J.M., Quak, C.W., Visser, T., Bakker, N.: A semantic rich Multi-Scale Information Model Topography. Accepted for publication in International journal of geographical information science, IJGIS (2010)
50. van Oosterom, P., Schenkelaars, V.: The Development of an Interactive Multi-Scale GIS. International Journal of Geographical Information Systems 9, 489–507 (1995)

51. van Oosterom, P.J.M.: Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest. Cartography and Geographic Information Science 32, 331–346 (2006)
52. Meijers, M., van Oosterom, P.J.M., Quak, C.W.: A storage and transfer efficient data structure for variable. In: Bernard, Sester, P. (eds.) Advances in GIScience, pp. 345–367. Springer, Heidelberg (2009)
53. Vermeij, M., van Oosterom, P., Quak, W., Tijssen, T.: Storing and using scale-less topological data efficiently in a client-server DBMS environment. In: 7th International Conference on GeoComputation, Southampton (2003)
54. Croom, F.H.: Principles of Topology. Cengage Learning, 312 (2002)
55. Cromwell, P.R.: Polyhedra, p. 460. Cambridge University Press, Cambridge (1999)
56. Coxeter, H.S.M.: Regular Polytopes, p. 321. Dover Publications, Mineola (1973)
57. Giblin, P.J.: Graphs, Surfaces and Homology: An Introduction to Algebraic Topology, 2nd edn. OGC 1999. Chapman and Hall, New York (1981)
58. OGC, OpenGIS Simple Features Specification For SQL, Revision 1.1 (1999), `http://www.opengeospatial.org/standards/sfs`

Author Index