Advances in Geographic Information Science

Sabine Timpf
Patrick Laube *Editors*

Advances in Spatial Data Handling

Geospatial Dynamics, Geosimulation and Exploratory Visualization

Springer

Advances in Geographic Information Science

For further volumes:
http://www.springer.com/series/7712

Sabine Timpf · Patrick Laube
Editors

Advances in Spatial Data Handling

Geospatial Dynamics, Geosimulation and Exploratory Visualization

Springer

Editors
Sabine Timpf
Department of Geography
University of Augsburg
Augsburg
Germany

Patrick Laube
Department of Geography
Geographic Information Science
University of Zurich—Irchel
Zurich
Switzerland

ISSN 1867-2434 ISSN 1867-2442 (electronic)
ISBN 978-3-642-32315-7 ISBN 978-3-642-32316-4 (eBook)
DOI 10.1007/978-3-642-32316-4
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012949373

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The International Symposium on Spatial Data Handling (SDH) is the biennial international research forum for Geographic Information Science (GIScience), co-organized by the Commission on Geographic Information Science and the Commission on Modeling Geographical Systems of the International Geographical Union (IGU). The conference brings together scholars and professionals from a wide range of disciplines, including (but not limited to) geography, computer science, information science, cognitive science, engineering, statistics and geostatistics, as well as from a rich spectrum of application sciences invaluably contributing to the development of the theory of GIScience.

With its beginning in 1984, the conference has a long tradition and evolved in parallel with the ever since developing discipline of GIScience. It still is a leading scientific meeting in the area. After the conference's third appearance in Australasia two years ago (held in Hong Kong, after Sydney 1988 and Beijing 2000), in 2012 SDH returned to Europe. The 2012 conference was held in Bonn, Germany, in conjunction with and prior to the 32nd International Geographical Congress (IGC), the quadrennial meeting of the International Geographical Union (IGU).

As a novelty, the 15th Symposium on Spatial Data Handling—SDH 2012 provided two different submission tracks, honoring different publication cultures: Full papers, approximating 15 pages, presenting in-depth research and short papers with a maximum of 6 pages highlighting current research, late-breaking results, and presenting concept papers. Twenty-eight full papers were submitted out of these 14 papers were selected for publication within this volume, resulting in a 50 % acceptance rate. The papers are ordered alphabetically by the first author. Short papers are published online with www.ceur-ws.org, with the option for extended papers to get published in a special issue of the International Journal of Geographical Information Science.

The research covered by the canon of papers in this volume allows for two interesting observations. First, all four fundamental GIS functions (1. data capture, 2. modeling, storage, and retrieval, 3. manipulation and analysis, and 4. presentation and sharing) keep on producing interesting and relevant contributions to the theory. Second, classic Spatial Data Handling topics (e.g., modeling topological relations, modeling terrain) are complemented with new and emerging challenges (spatio-temporal and mobile GIS, new geodata sources), mirroring the progress of the discipline.

A first cluster of papers explores new forms of capturing geodata. A diverse tool-set is used for searching and structuring geoinformation from a wide range of structured and unstructured data sources, online and offline. The work by Qi and Yeh presents an integrated approach for land-use classification from remote sensing data. The authors combine object-oriented image analysis with decision trees and support vector machines, reviving this spatial data analysis classic with data mining and machine learning techniques. Walter et al. address the important challenge of accessing map data on the web. Their approach combining web crawling with an automatic interpretation of the map type based on artificial neural networks shows promising results for retrieved vector maps. The paper by Tsatcha et al. introduces an algorithm for retrieving semantic information from a corpus of traffic regulations for maritime navigation using geographic information retrieval combined with natural language processing.

A second cluster of papers investigates novel approaches for modeling a dynamic world, revisiting topological relations, and terrain modeling, as well as integrating the fourth, i.e., the temporal dimension. The paper by Stuetzle and Franklin describes a novel operation called the drill, which is able to represent and regenerate terrains. The operation was developed as an alternative to other terrain representations such as TINs and has the advantage that it retains drainage information during encoding. The paper by Guerra-Filho, Medeiros, and de Rezende presents a new model for describing topological relations among geometric objects based on intersection matrices representations (simplicial complexes). This is an extension of the classical boundary/interior/exterior model by expressing and counting the ways in which two geometric objects are spatially related and then storing this information using the third dimension. Putting topology into practice, Wallgrün presents a method for automatically resolving topological inconsistencies between polygons, e.g., fixing a gap between a bridge and the respective shoreline. His approach is based on formalizing the topological constraints and then transforming the formalization into an optimization problem. Breunig et al. present a service-based geo-database architecture designed to handle geospatial and time-dependent data. The potential of their architecture is illustrated through the modeling of complex surface and volume changes in an open-pit mine. Hong et al. develop a location-aware push service that provides timely messages of campus events to students by taking both the spatial and the temporal perspectives into account. The focus in this study is on the constraints on the temporal information as well as on the space–time location of the recipient of a message.

Aiming for a better understanding of our natural and built environment, spatial analysis remains a stronghold of the conference. The volume features innovative techniques and algorithms for structuring space, as well as analyzing biological and socio-economic variables. Buchin et al. investigate the segmentation of movement trajectories. Their segmentation shall divide GPS tracks of migrating geese into semantically homogeneous units, aiming at the identification of animal behaviors such as flying, resting, or foraging. Guerra-Filho and Samet introduce a hybrid shortest path algorithm to perform intra-regional queries used e.g., in moving object databases. Regions are formed by subdividing a single network into smaller regions creating a hierarchy of multilevel networks. The shortest path is determined in each region and then concatenated using their new method. Shannon and Harvey found that the usability of data on food assistance benefits can be increased when using a three-class areal interpolation method to disaggregate the data. They show that their interpolation method and subsequent spatial analysis allows for a more accurate depiction of how residents actually access the food system.

Finally, a fourth group of papers addresses how spatial data could and should be presented and communicated, addressing visualization and cognitive aspects. Suleiman et al. developed a new algorithm for isovist calculation in a 2D and in a 3D environment. The algorithm calculates a 3D isovist in an urban environment based on a DEM. The method allows for the calculation of a higher variety of visibility indices than existing algorithms. Gousie presents a tool for visualizing data quality properties of digital elevation models. The 3D tool set allows the display of the DEM together with its uncertainty and error, keeping both local and global scales in perspective. Klippel et al. evaluate the cognitive adequacy of the DLine-Region calculus by presenting animated icons to participants of a behavioral study. They report that some topological relations form conceptual clusters but these findings do not correspond to assertions in the literature. In general more behavioral studies are needed to prove the claim that qualitative calculi are cognitively more adequate than other calculi.

We thank the many people who made SDH 2012 possible: all those who submitted their research and participated in the meeting, the program committee for reviewing and sharing their experience, the steering committee for their support, the local organizing committee, the staff of the GSI, and especially Brian Lees for encouragement.

June 2012

Sabine Timpf
Patrick Laube

Committee

Steering Committee

Sabine Timpf, University of Augsburg, Germany (General chair)
Anne Ruas, IGN, France
Monika Sester, University of Hannover, Germany
Itzhak Benenson, Tel Aviv University, Israel
Stewart Fotheringham, University of St Andrews, UK
Klaus Greve, University of Bonn, Germany
Bin Jiang, University of Gävle, Sweden
Patrick Laube, University of Zurich, Switzerland
Stephan Winter, University of Melbourne, Australia

Program Committee

Torun Abdülvahit
Masatoshi Arikawa
Bénédicte Bucher
Christophe Claramunt
Eliseo Clementini
Leila De Floriani
Hande Demirel
Thomas Devogele
Peter Fisher
Manfred Fischer
W. Randolph Franklin
Mark Gillings
Michael Goodchild
Hans W. Guesgen
Francis Harvey
Christopher Jones
Marinos Kavouras
Brian Klinkenberg
Brian Lees
Michael Leitner
Lily Chao Li
Zhilin Li
Yaolin Liu
William Mackaness
Hervé Martin
Liqiu Meng
Martien Molenaar
Sebastien Mustière
Donna Peuquet
Ross Purves
Juri Roosaare
Yukio Sadahiro
Nicholas Tate
Vit Vozenilek
Monica Wachowicz
Shuliang Wang
Robert Weibel
Qihao Weng
Michael Worboys
Tang Xinming
Anthony Yeh
Sisi Zlatanova

Organizing Committee

Sabine Timpf, University of Augsburg, Germany
Carolin von Groote-Bidlingmaier, University of Augsburg, Germany
David Jonietz, University of Augsburg, Germany
Petra Richter (finances), University of Augsburg, Germany
Tobias Michl (conference secretary), University of Augsburg, Germany
Esther Owald (conference secretary), University of Augsburg, Germany
Nicole Karrais (web page design), University of Augsburg, Germany
Hartmuth Basan (logo and poster design), University of Augsburg, Germany

Contents

Topological and Geometric Data Handling for Time-Dependent Geo-Objects Realized in DB4GeO

Martin Breunig, Edgar Butwilowski, Paul Vincent Kuper, Daria Golovko and Andreas Thomsen

Abstract In advanced spatio-temporal scenarios, such as the simulation of complex geo-processes, the analysis of complex surface- and volume-based objects changing their locations and shapes in time is a central task. For example, the documentation of landfills, mass movements or volcanic activities requires 4D modeling based on dynamic geometric and topological database structures. In this contribution we present our concepts and implementation efforts for the effective handling of geospatial and time-dependent data realized in DB4GeO, a service-based geo-database architecture. The topological and geometric data models of DB4GeO are described in detail. A geoscientific application of an open-pit mine demonstrates the usefulness of the concepts introduced at the beginning of the paper. Finally, an outlook is given on future geo-database work dealing with extensions of DB4GeO and the handling of geo-objects in the context of cooperative 4D metro tracks planning

M. Breunig (✉) · E. Butwilowski · P. V. Kuper · D. Golovko
Geodetic Institute, Karlsruhe Institute of Technology, Englerstr.7,
76131 Karlsruhe, Germany
e-mail: martin.breunig@kit.edu

E. Butwilowski
e-mail: edgar.butwilowski@kit.edu

P. V. Kuper
e-mail: kuper@kit.edu

D. Golovko
e-mail: daria.golovko@kit.edu

A. Thomsen
Institute of Geosciences, Christian-Albrechts-Universität zu Kiel, Otto-Hahn-Platz 1,
24118 Kiel, Germany
e-mail: athomsen@geophysik.uni-kiel.de

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_1,

1 Introduction

The spatial data handling community looks back on a history of more than 25 years (Goodchild 1990; Marble DF 1984). During this time the geospatial data handling of volumetric 3D objects and their operations have been examined under different aspects by (Balovnev et al. 2004; Breunig 2001; Breunig et al. 1994; Döner et al. 2010; Mäntylä 1988; Pigot 1992; Pouliot et al. 2007, 2008, 2010; Schaeben 2003) and by other authors.

Based on this tradition and on the experiences of GeoToolKit, an object-oriented geo-database kernel for the management of 3D geometries (Balovnev et al. 2004), we have developed a geo-database architecture called DB4GeO (Bär 2007; Breunig et al. 2010). Right from the beginning, DB4GeO has been designed to support advanced geo-scenarios that require a web-based 3D/4D data access. It is implemented completely in the Java programming language and is based on the free object-oriented database management system db4objects (Versant 2011). The services of DB4GeO can be divided into primitive and complex services (Breunig et al. 2010). The primitive services contain basic geometric and topological operations such as calculating the distance between two objects, their relative position to each other (distinct, meet, overlap etc.), and the computation of the intersection between two objects. An example for a complex service is the "3D–2D service" that computes a profile section—i.e. the intersection of a vertical plane with 3D geometries—within a geological block model (Breunig et al. 2010). At the moment, the service architecture is implemented using REST (Fielding and Taylor 2002).

The paper is organized as follows. Sections 2 and 3 are dedicated to the concepts and implementations for spatio-temporal data handling in DB4GeO focusing on topological and geometric database support. In Sect. 4, a geoscientific application is presented. Finally, Sect. 5 gives an outlook on our future work.

2 Spatio-Temporal Concepts of DB4GeO

2.1 Geometric and Topological Core Model

The DB4GeO core API implements the simplicial complex model for the spatial part of its 3D object model.[1] The core API defines a 3D object to be an object in 3D space that has a spatial part which can be a *sample*, a *curve*, a *surface* or a *volume*. These abstract geometry concepts are specified by specific geo-objects as follows[2]:

[1] For an elaborate UML class diagram of the DB4GeO kernel geo-object model cf. Bär (2007), p. 65.

[2] For a visual overview of the geometry model of DB4GeO, cf. Butwilowski and Breunig (2010).

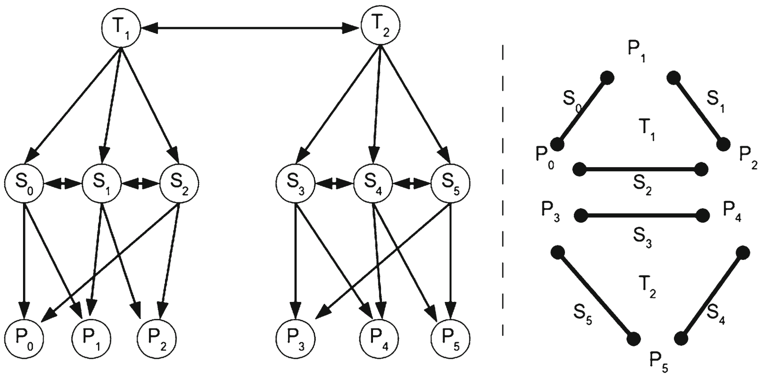


Fig. 1 Part of the incidence graph for simplicial complex model of the DB4GeO kernel

- sample as *point net*,
- curve as *segment net*,
- surface as *triangle net*, and
- volume as *tetrahedron net*.

The aforementioned nets are subdivided into disconnected *net components*. A net component itself consists of connected simplices (also *simple geo-object*). By means of the core API it is possible to navigate on top of net components by iterating over the explicitly stored *contact relations* between the simplices of the net.[3] Figure 1 shows a part of the incidence graph of the simplicial complex model as it is implemented in DB4GeO.

The arrows (left side) represent the connections between the simplices. Depicted is an example of two triangles T_1, T_2 that are adjacent by the segments S_2, S_3 (see right side). There are directed top-down incidence relations *from triangles to segments to points* as well as "next to" connections between multiple triangles and between multiple segments. This incidence graph is commonly used in geometry modeling systems (Lévy and Mallet, 1999, cf. p. 3), but obviously there are also some insufficiencies concerning the navigational properties of this structure. For example, there are no back references from lower to higher dimension simplices as well as there is e.g. no direct connection between S_2 and S_3, which makes navigation quite difficult. In some cases, it is necessary to traverse the whole structure to do one step in navigation.[4] While this model is suitable for many applications, it also has some shortcomings. For example, it is not possible to distinguish *subdivisions* on a net component, i.e. cells (e.g. faces/volumes) composed of multiple simple geo-objects (triangles/tetrahedrons) forming a net component. A potential improvement of this shortcoming will be discussed in Sect. 2.3.

[3] This structure can be seen as the implicit topology model of the DB4GeO/DB3D core API.

[4] For example, if it is necessary to find all neighboring segments to a given point.

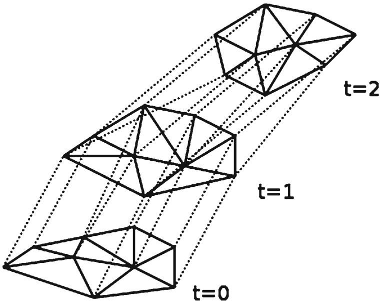


Fig. 2 Representation of a 4D object (Rolfs 2005)

2.2 A First Geometric 4D Model

To support spatio-temporal data handling in DB4GeO, a 4D model on top of the described geometric core model has been developed (Rolfs 2005). Due to this first 4D model, DB4GeO was able to handle simplex-based data within a fixed time interval. Such a 4D object has a spatial part which can be a sample, a curve, a surface or a volume (cf. 3D object) and is located in a 4D space. Figure 2 shows the representation of such a 4D object, which is comparable to the representation of Worboys's spatio-temporal model (Worboys 1992).

Due to the internal implementation, the first geometric 4D model of DB4GeO is not able to extend the time interval of an existing 4D object with further data. Furthermore, the data needs to meet several constraints to work properly. The major constraints are:

- The import format is proprietary;
- The net topology cannot change within the time interval;
- The sequence of time intervals is fixed and cannot be modified.

According to these constraints a new 4D model to handle spatio-temporal data was developed in Kuper (2010). The techniques to improve the functionality, user experience, and performance are described in Sects. 2.4 and 3.2.

2.3 Topological Structure

For the construction of regions by aggregation of multiple triangles or tetrahedra, following a naive approach, it would be sufficient to assign to every individual triangle or tetrahedron of the simplicial complex an attribute that determines to which region the respective simplex belongs. However, with this naive approach it would not be possible e.g. to navigate along the edge geometries of the regions (for example along the boundary surface between two volumes or along the boundary segment between two surfaces) efficiently. That is why DB4GeO handles topology in a more generic way following works of the 3D modeling community by Lienhardt (1989), Brisson (1989), and Lévy and Mallet (1999).

Brisson and Lienhardt proposed explicit generic topology models that address *cell-tuple structures* and *Generalized Maps*, respectively. The way to the cell-tuple structures and the Generalized Maps was paved by prior topological models that have widely been used in the CAD community, such as the *winged edge representation* of Baumgart and the *half-edge structure* and *radial edge representation* of Weiler (Baumgart 1975; Weiler 1985, 1986). The cell-tuple structure has finally been proposed by Brisson (Brisson 1989).

Brisson introduced the notion of *cell-tuple* that is defined as an ordered sequence of cells of decreasing dimension: a cell-tuple corresponds to a path in the incidence graph (cf. Fig. 1). The cell-tuples are "connected" through the concept of adjacency that is inherent to the cell-tuple structure: two cell-tuples C and C' are called i-adjacent (A_i) if exactly one cell, namely the cell of dimension i of the cell-tuple is exchanged (so called *switch* operation) so that another tuple of the set of valid cell-tuples (a *permutation*) is obtained in return.

Lienhardt proposed the d-Generalized Map, (d-G-Map), a more abstract model of the topology of a d-dimensional cellular complex, based on algebraic topology (Lienhardt 1989). A *d-G-Map* is defined as a pair of a set of *darts* and of a set of $d+1$ *involutions*, noted α_i, i.e. transformations defined on the darts verifying $\alpha_i\alpha_i = id$ for $i = 0, 1, \ldots d$. Moreover, for any pair of indexes i, j with $j = i + 2 + k$, $\beta_{ij} = \alpha_i\alpha_j$ again is an involution, which implies that α_i and α_j commute. With this structure, a d-G-Map can be interpreted as a special *abstract simplicial complex*. A possible representation of a d-G-Map is a d-celltuple structure. Another possible representation is a graph *G(D,A)* with the set D of darts as nodes, and the set A of edges composed of d classes of pairs of darts defined by the $d + 1$ involutions $\alpha_0 \ldots \alpha_d$.

A particular class of G-Maps are *orientable d-G-Maps*. These can be represented by bipartite graphs with two classes of darts with opposite "polarity", linked by the edges defined by the α_i involutions. For our practical applications in geosciences, only orientable G-Maps are considered. A spatial model of a d-G-Map (or of a celltuple structure) is obtained by an *immersion* into the euclidean space R^d. Thus different spatial models can be derived from the same G-Map by applying different *immersions*.

2.4 Advanced Geometric 4D Model

The new model to handle geometric spatio-temporal data in DB4GeO improving the first model introduced in Sect. 2.2 is based on three main techniques with the following objectives:

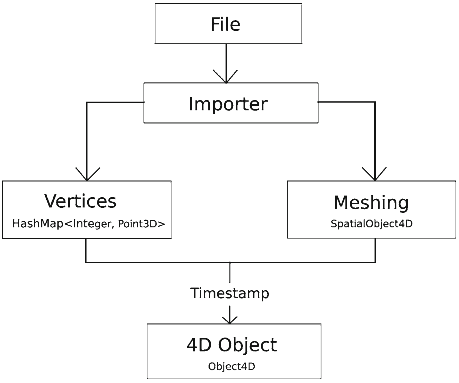


Fig. 3 Workflow adding a time step to a 4D object in DB4GeO (Kuper 2010)

- PointTubes: to handle and store the points of a time-dependent 2–3D simplex net efficiently;
- Pre- and post-objects (Polthier and Rumpf 1995): to offer a solution for changing net topology within time;
- Delta-storage (Strathoff 1999): to avoid redundant storage of points and topology information.

The spatio-temporal handling of data is focused on moving vertices in a 3D space within a time interval. DB4GeO handles the information about the net topology separately from the vertices. Due to the implementation of the spatio-temporal model presented in Polthier and Rumpf (1995), the net topology can change in time. At every time step, a pre-object and a post-object exist. The pre-object of time step t_i and the post-object of time step t_{i-1} have the same net topologies, i.e. discretizations. The pre-object and the post-object of one time step have the same geometry, i.e. the location of the vertices, while their net topologies can differ. The discretization can change at every time step t_i while the geometry can change between two time steps.

DB4GeO provides two main functions to build a proper 4D object: *addTimestep()* and *addTopology*. The former function adds the information about the vertices while the latter adds the net topology. The topological information about the net structure is added to the 4D object with every change of the topology, i.e. in a time step with a pre- and post-object. Therefore, DB4GeO is able to interpolate between time steps without paying attention to the net topology. It interpolates the vertices stored in PointTubes and uses the applicable topology when needed.

The time interval is extensible, i.e. it is possible to add additional time steps to an existing 4D object with a simultaneous update of the end date of the time interval. To improve the performance and to reduce the amount of storage data, DB4GeO only stores those objects of every new time step which contain new information, i.e. only the changes are stored. Every insertion of a new time step compares the vertices to the last one. Figure 3 provides an overview of the new 4D model workflow.

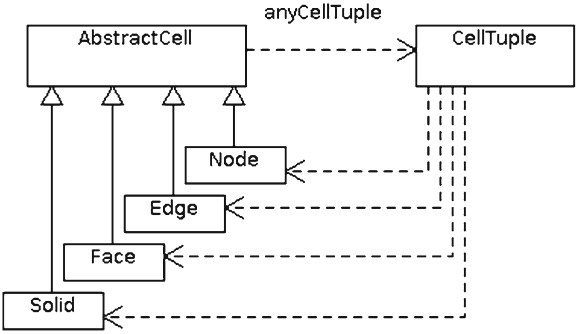


Fig. 4 References between abstract cell, cells and cell-tuples

3 Implementation of the Spatio-Temporal Concepts

3.1 Implementation of the Topology Concept

The G-Maps module for the management of the topology for 3D geo-objects in DB4GeO conceptionally relies on the notion of cells as a means to describe a geo-object by its decomposition. In the context of DB4GeO a cell may be any object that is composed of a simply connected set of simple geo-objects, e.g. a curved surface or a polyhedron or a solid. From a software modeling perspective, the conjunction between the classes of simple geo-objects, as they are implemented in the DB4GeO core API, and the cell classes, as provided by the G-Maps module (i.e. the conjunction between DB4GeO and the G-Maps module), are depicted and described in Butwilowski and Breunig (2010). Thus, the next step is to enable a connection of the cell classes to a cell-tuple class. This model is represented in Fig. 4, where the topological cell classes *Node*, *Edge*, *Face* and *Solid* (these are supported by the G-Maps module) are generalized by an `AbstractCell` class that has a reference `anyCellTuple` to a `CellTuple` class. Since all cell classes are of type `AbstractCell`, any cell "has a" cell-tuple. This cell-tuple represents an arbitrary cell-tuple of the cell. Conversely any cell-tuple has separate references to all four cells (cf. Fig. 4). This allows an unfettered back-and-forth navigation between the objects of all the cell types and the respective cell-tuple objects.

The class diagram considers the definitions of cell-tuple structure and G-Maps of Brisson and Lienhardt and serves as a basis for the innermost kernel of the G-Maps module for DB4GeO. An object of the `CellTuple` class is a composition of `Cell` objects of different dimensions[5] (which represent the incident cells). The field of each `CellTuple` object includes references to exactly four `CellTuple` objects

[5] i.e. of objects of the classes *Node*, *Edge*, *Face* and *Solid*.

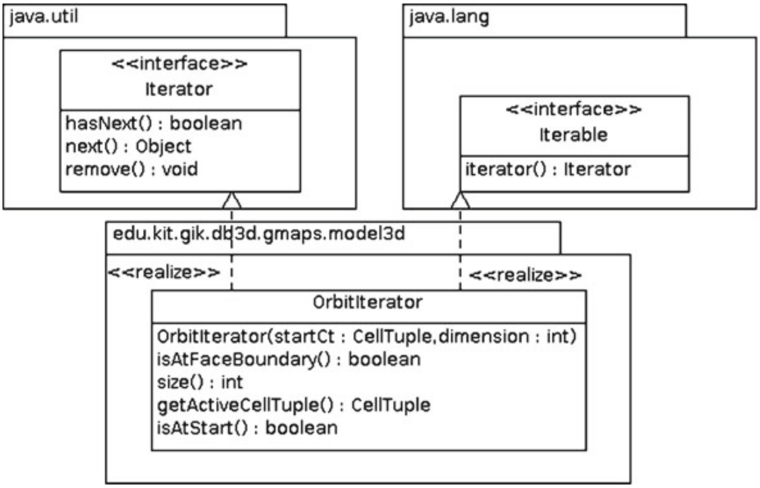


Fig. 5 Diagram of *OrbitIterator* class

called `alpha0`, `alpha1`, `alpha2`, and `alpha3`.[6] These references provide the means to perform fundamental/basic topological operations to navigate between the cells in any direction (these are the above mentioned G-Map operations).

These fundamental operations can be combined to more complex topological operations that are capable of traversing whole cells (so called *orbits*, cf. (Lévy and Mallet 1999, p. 5)). Orbits fit well with the concept of *iterators* (key concept of the Java programming language) which are defined by the interface `java.lang.Iterator`.[7] In our implementation, an orbit is represented by an `OrbitIterator` class that realizes the `Iterator` and the `Iterable` interfaces of Java (cf. Fig. 5).

Thus an `OrbitIterator` provides itself through its `iterator()` method and may directly be used in a `for`-loop. As a realisation of the `Iterator` interface, the `OrbitIterator` provides the methods `hasNext()` and `next()`.[8] For the instantiation of an `OrbitIterator` object, its constructor needs an object of type `CellTuple` (as its constructor parameters) which will be the start cell-tuple (*startCt*) of the orbit (this can be any cell-tuple of a cell in fact) and an integer value that defines the *dimension* of the orbit (cf. Fig. 5). As a result, a complete 0-dimensional orbit traversal around the node with dart `startCt` (i.e. the traversal of an orbit that would "collect" all cell-tuples around that node) is as simple and as elegant in our Java implementation as in the following example:

[6] These are not illustrated in Fig. 5 to reduce the diagram's complexety.

[7] Though, to be more precise, orbits are not only iterators but *circulators* (Devillers et al. 2011).

[8] The `OrbitIterator` does not provide a `remove()` method (in conformity with the orbit definition).

```
for(CellTuple ct : new OrbitIterator(startCt, 0))
{
  System.out.println(ct);
}
```

In our implementation, all connected cells of a cell net are summarized in a cell net component. Every cell net component is further subdivided into two *cell net levels*, i.e. a cell net component at *network level* or just *net level* (C_{NL}) and a cell net component at *object level* (C_{OL}). The topology of C_{NL} is an exact reproduction of the topology of the net structure of the underlying simple geo-objects net (i.e. of the simplicial complex). This level is mainly used for navigation purposes. It eases the algorithmic navigation on the net structure. C_{OL} on the other hand is the boundary representation of the component object (i.e. representing the whole geo-object itself). The topology defined by the cell-tuple structure cannot be edited by the user on C_{NL}, only on C_{OL}.

3.2 Implementation of the Geometry Concept for Time-Dependent Objects

For an efficient and user-friendly behavior of our 4D model we developed a simple API to work with 4D objects in DB4GeO. The main concepts mentioned in Sect. 2.4 are implemented in the class *object4D*. All three concepts work automatically. The implementation is based on two main functions:

addTimestep: the function is called with two parameters, the *vertices* which extend or create the PointTubes of an 4D object and a *java.util.Date* object to specify the time stamp. If there are any vertices already existing in the last time step, the internal PointTubes will be extended with references to these. Otherwise new *Point3D* objects for the extension of our PointTubes are created.
addTopology: the function is called with one parameter, representing the net topology of the just added time step. This object is called *SpatialObject4D* and consists of 0–n *Point4DNet*, *Segment4DNet*, *Triangle4DNet* or *Tetrahedron4DNet* objects. This information will be added to the 4D object. The number of *SpatialObject4D* objects will only increase if the last time step was a post-object, i.e. the *addTimestep* function was called twice with the same *Date* object.

To access single states of 4D objects at arbitrary dates we developed a class called *ServicesFor4DObjects*. This class contains one main function:

***getInstanceAt(Object4D, Date)*:** this functions creates a 3D object, i.e. a snapshot of the 4D object at the specified date via interpolation. The date must be part of the time interval. Due to the use of the implemented spatio-temporal model introduced in Polthier and Rumpf (1995), the computation of such a snapshot always refers to an interpolation between two sets of vertices in a *1:1* relation.

4 A Geoscientific Application Example

One of the use cases for DB4GeO has been the so-called Piesberg application. Piesberg is a hill near the city of Osnabrück in Northern Germany that has been exposed to open-pit mining for several decades. In the 1970s, its older part began to be used as a landfill causing changes to the volume and surface of the hill. A dataset representing the surface changes of Piesberg in 12 time steps between 1976 and 1993 is available (Lautenbach and Berlekamp 2002).

The present application example demonstrates how the topology module of DB4GeO can be used for a land use classification. Different classes of land use on Piesberg can be identified dividing the surface of the hill into five regions: "mining" (has a hole), "wind energy", "old landfill", "active landfill", "compost". Managing non-simplex regions necessary for this kind of applications would not have been possible using the geometry model of DB4GeO alone.

The G-Maps module of DB4GeO offers four operations for editing the composition of a face net: `insert node`, `remove node`, `insert edge`, and `remove edge`. The `insert edge` operation enables inserting both simple edges (with only two endpoints) and multi-edges (with more than two vertices). Before each of the operations is carried out, constraints are checked to avoid an invalid result. Examples of insertion and removal operations are shown in Fig. 6.

To simulate the Piesberg landfill, at first a face net with a net level and an object level is created. The net level represents the geometry of the object, in this case via a triangle net. The object level initially has one face; its boundary coincides with the boundary of the whole object. Then, we used the insertion and removal operations implemented in DB4GeO to divide the surface of the hill into the five land use classes. Notice that one of the five resulting regions of the object level has a hole. Finally, we used the `OrbitIterator` (cf. Sect. 3.1) to find faces of the net level that correspond to each face of the object level. Since the faces of the net level are simplices, we easily exported them from our geo-database into the gOcad® format (.ts) and visualized them in ParaViewGeo® (cf. Fig. 6).

Our next research goal is to extend the G-Maps module of DB4GeO to handle temporal changes and test it with all of the 12 available datasets of Piesberg.

5 Conclusion and Outlook

In this contribution we have presented concepts and implementations for geospatial and time-dependent data handling of complex geometric and topological objects realized in DB4GeO, our service-based geo-database architecture. Advanced topological and time-dependent geometric data models have been introduced. In our future work we will also focus on the straight-forward visualization of database results via WebGL from standard Internet browsers. Furthermore, the management

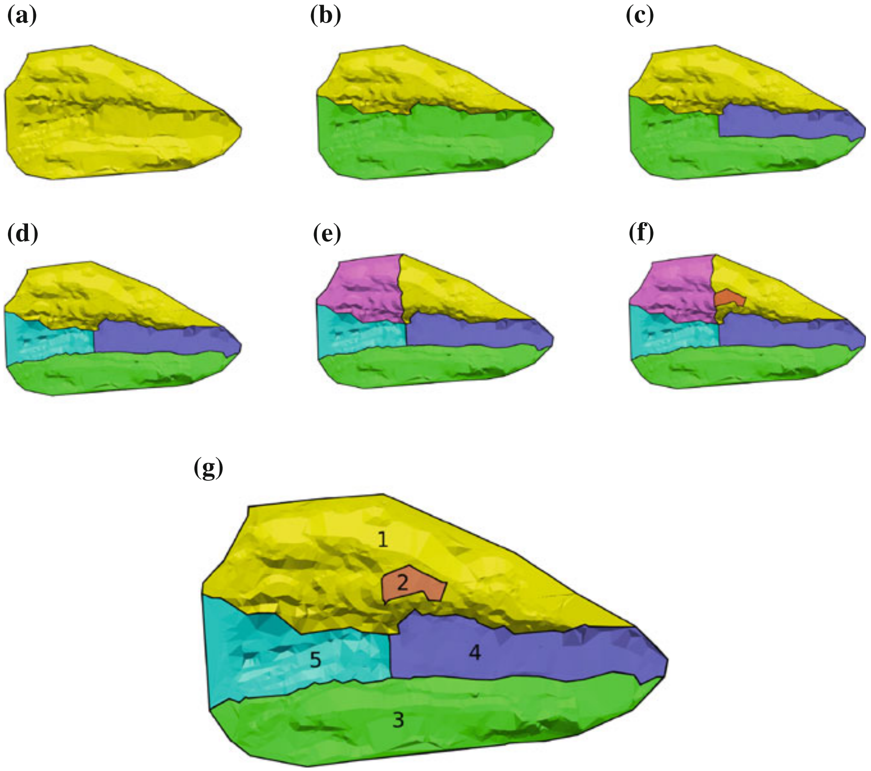


Fig. 6 Insertion and removal operations on Piesberg dataset (visualized with ParaviewGeo®) **a** Befor editing. **b** Edge inserted. **c** Edge inserted. **d** Edge inserted. **e** Edge inserted. **f** Two nodes and an edge inserted. **g** Two edges removed. Land use classes : 1- mining, 2- wind energy, 3- old landfill, 4- active landfill, 5- compost

of city models in DB4GeO will be examined using GML data. Another open question is how DB4GeO can be adapted to OGC Web services. Finally, we intend to examine the handling of 3D geo-objects used for the cooperative planning of metro tracks. Therefore, spatial representations other than simplicial complexes such as boundary representation or parameterized geometries should be directly supported by the geo-database.

Acknowledgments We thank Jürgen Berlekamp from USF, Osnabrück University and the Survey Office of Osnabrück city for the permission to use the Piesberg dataset for scientific purposes.This research has been funded by the German Research Foundation (DFG), grant no. BR2128/12-1 and BR2128/14-1.

References

Balovnev O, Bode T, Breunig M, Cremers AB, Möller W, Pogodaev G, Shumilov S, Siebeck J, Siehl A, Thomsen A (2004) The story of the geotoolkit–an object-oriented geodatabase kernel system. GeoInformatica 8(1):5–47

Bär W (2007) Verwaltung geowissenschaftlicher 3D daten in mobilen datenbanksystemen. Ph.D thesis, University of Osnabrück, Germany

Baumgart BG (1975) A polyhedron representation for computer vision. In: Proceedings of 1975 national computer conference, AFIPS, pp 589–596, 19–22 May 1975

Breunig M (2001) On the way to component-based 3D/4D geoinformation systems. Lecture notes in earth sciences, vol 94. Springer, Heidelberg, p 199

Breunig M, Bode T, Cremers AB, (1994) Implementation of elementary geometric database operations for 3D-GIS. In: Waugh T, Healey R (eds) Proceedings of the 6th international symposium on spatial data handling, Edinburgh, pp 604–617

Breunig M, Schilberg B, Thomsen A, Kuper PV, Jahn M, Butwilowski E (2010) DB4GeO, a 3D/4D geodatabase and its application for the analysis of landslides. Lecture notes geoinformation and cartography risk crisis management, pp 83–102

Brisson E (1989) Representing geometric structures in d dimensions: topology and order. In: Proceedings of the 5th ACM symposium on computational geometry, ACM Press, Washington, pp 218–227

Butwilowski E, Breunig M (2010) Requirements and implementation issues of a topology component in 3D geo-databases. Poster Abstract. In: Proceedings of the 13th AGILE international conference on geographic information science

Devillers O, Kettner L, Pion S, Seel M, Yvinec M (2011) Handles, ranges and circulators, September 2011. http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Circulator/Chapter_main.html. Accessed 06 Jan 2012

Döner F, Thompson R, Stoter J, Lemmen C, Ploeger H, van Oosterom P, Zlatanova S (2010) 4d cadastres: first analysis of legal, organizational and technical impact–with a case study on utility networks. Land Use Policy 27:1068–1081

Fielding RT, Taylor RN (2002) Principled design of the modern web architecture. ACM Trans Internet Technol 2(2):115–150

Goodchild MF (1990) Keynote address: spatial information science. In: Proceedings, fourth international symposium on spatial data handling, Vol 1. Zurich, pp 13–14

Kuper PV (2010) Entwicklung einer 4D objekt-verwaltung für die geodatenbank DB4GeO. University of Osnabrück, Germany, Diploma thesis

Lautenbach S, Berlekamp J (2002) Data set for the visualization of Piesberg Landfill in Osnabrück. University of Osnabrück, Germany Institute of Environmental Systems Research

Lévy B, Mallet J-L (1999) Cellular modeling in arbitrary dimension using generalized maps. http://www.alice.loria.fr/publications/papers/1999/g_maps/g_maps.pdf. Accessed 14 Dec 2011

Lienhardt P (1989) Subdivisions of n-dimensional spaces and n-dimensional generalized maps. Proceedings of the fifth annual symposium on computational geometry, Washington, ACM Press, In, pp 228–236

Mäntylä M (1988) Introduction to solid modeling. Computer Science Press, Rockville

Marble DF (ed) (1984) In: Proceedings, international symposium on spatial data handling, Zürich (1984) Geographisches Institut. Universität Zürich-Irchel, Abteilung Kartographie/EDV

Pigot S (1992) A topological modeling for a 3D spatial information system. In: Proceedings of the 5th international symposium on spatial data handling, Charleston, South Carolina, pp 344–360

Polthier K, Rumpf M (1995) A concept for time-dependent processes. Visualization in Scientific Computing, Springer, Berlin pp 137–153

Pouliot J, Badard T, Desgagné E, Bédard K, Thomas K (2007) Development of a web geological feature server (WGFS) for sharing and querying of 3D objects. In: van Oosterom et al. (eds) Advances in 3D geoinformation systems–Lecture notes in geoinformation and cartography, pp 115–130

Pouliot J, Bédard K, Kirkwood D, Lachance B (2008) Reasoning about geological space: coupling 3D geomodels and topological queries as an aid to spatial data selection. Comput Geosci 34(5):529–541

Pouliot J, Roy T, Fouquet-Asselin G, Desgroseilliers J (2010) 3D Cadastre in the province of quebec: a first experiment for the construction of a volumetric representation. In: Kolbe GNC, Knig TH (eds) Advances in 3D geo-information sciences. Lecture notes geoinformation and cartography

Rolfs C (2005) Konzeption und implementierung eines datenmodells zur verwaltung von zeitabhängigen 3D-modellen in geowissenschaftlichen anwendungen. University of Osnabrück, Germany, Diploma thesis

Schaeben H, Apel M, Boogaart KGvd, Kroner U, (2003) GIS 2D, 3D, 4D, nD. Von geographischen zu geowissenschaftlichen informationssystemen. Informatik-Spektrum 26(3):173–179

Strathoff F (1999) Speichereffiziente verwaltung zeitabhängiger geometrien für ein geotoolkit. University of Bonn, Germany, Diploma thesis

Versant (2011) http://www.db4o.com. Accessed 22 Nov 2011

Weiler K (1985) Edge-based data structures for solid modeling in curved-surface environments. Comput Graph Appl 5(1):21–40

Weiler K (1986) The radial edge structure: a topological representation for non-manifold geometric boundary modeling. In: Proceedings of the IFIG WG 5.2

Worboys MF (1992) A model for spatio-temporal information. In: Proceedings of the 5th international symposium on spatial data handling. Vol 1, pp 602–611

Segmenting Trajectories by Movement States

Maike Buchin, Helmut Kruckenberg and Andrea Kölzsch

Abstract Dividing movement trajectories according to different movement states of animals has become a challenge in movement ecology, as well as in algorithm development. In this study, we revisit and extend a framework for trajectory segmentation based on spatio-temporal criteria for this purpose. We adapt and extend the framework to the setting of segmentation according to the individual movement states of an object, in particular an animal. We implement the framework and evaluate it at the example of tracks of migrating geese.

Keywords Trajectory segmentation · Experimental evaluation

1 Introduction

Due to advances in tracking technology, like GPS, a growing amount of trajectory data are being collected. Trajectory data occur in many contexts, such as sports, biology, traffic analysis, and defense. In particular, they occur in movement

MB is supported by the Netherlands Organisation for Scientific Research NWO (project ACATA) and partially supported by EU Cost action IC0903 (MOVE). AK is supported by the Strategic Fund of the KNAW, granted to Johan van de Koppel, Marc Naguib and Bart A. Nolet.

M. Buchin (✉)
Department of Mathematics and Computer Science, TU, Eindhoven, The Netherlands
e-mail: m.e.buchin@tue.nl

H. Kruckenberg
European White-Fronted Goose Research Programme, Verden, Germany

A. Kölzsch
Department of Animal Ecology and Project Group Movement Ecology,
Netherlands Institute of Ecology (NIOO-KNAW), Wageningen, The Netherlands

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_2,

ecology (Nathan et al. 2008): biologists use GPS tracks of animals to aid in the understanding of their movement. To analyze large amounts of trajectory data, efficient and effective methods are needed. *Automatic Movement Analysis* has emerged as a research field addressing this challenge (Shamoun-Baranes et al. 2012).

One important task is *trajectory segmentation*: cutting the trajectory into smaller pieces. This may have different purposes, e.g., data handling (efficient storing and searching) or analysis (learning about the underlying movement). We focus on the latter, and in particular on segmentation based on individual movement states of the object (animal). That is, we are given a trajectory and a set of classes of movement states, and we want to segment the trajectory according to these movement states. For instance, we want to segment a bird trajectory such that the bird was flying, foraging, or resting on each piece. More generally, this is a segmentation based on the *semantics* of the trajectory.

Recently, Buchin et al. (2011) proposed a framework for trajectory segmentation based on spatio-temporal criteria. We revisit this framework and extend it for the purpose of trajectory segmentation based on the individual movement states of an object. As a prototype example, we consider segmenting the trajectories of migrating geese.

Our framework is semi-automatic in the sense that the parameters for segmentation need to be input manually, and the resulting segmentation is then computed automatically. Other approaches (e.g. Fauchald and Tveraa 2003; van Moorter et al. 2010) are fully automatic, that is, no user input is required. We see the strength of our approach in that expert knowledge can be used to effectively describe movement states. We will discuss this further in Sect. 4.

Results. We extend and implement the framework presented in Buchin et al. (2011) for segmentation based on movement states of an object. In particular, we

- identify and discuss relevant criteria,
- adapt the framework for describing movement states,
- integrate a time criterion,
- allow outliers in the data,
- present the first implementation and experimental evaluation of the framework.

2 Extending the Framework

2.1 Existing Framework

The existing framework (Buchin et al. 2011) segments a trajectory such that each segment is homogenous in the sense that a set of spatio-temporal criteria are fulfilled. For this, it considers spatio-temporal *attributes*, such as heading, speed, and location. It uses *criteria* on the attributes, such as bounding the heading angular range, bounding the speed ratio, or requiring the locations to lie in a disk of given radius. The criteria are required to be monotone in the sense that if a segment fulfills a criterion, then so does every subsegment of the segment. Criteria can be combined, by a boolean or a linear combination. We call the result a *set of criteria*. The framework provides algorithms for segmenting a trajectory into the fewest number of pieces such that each piece fulfills the set of criteria.

The algorithms are greedy strategies: *incremental-search* and *double-and-search*. These greedy strategies can be described as follows. The algorithm starts at the beginning of the trajectory and finds the longest segment that fulfills the set of criteria. Then it starts new at the end point of the segment just found, and again finds the longest segment fulfilling the criteria. And so on. The incremental-search finds the longest segment by incrementing the test segment by one in each step. Double-and-search uses first an exponential search on the segment size, until this fails, and then a binary search between the last two points (the last where it succeeded and the first where it failed). The run time of both algorithms depends on the chosen criterion. In most cases, an overall $O(n \log n)$ run time is achieved, where n is the number of edges of the trajectory to be segmented (Buchin et al. 2011).

2.2 Analysis of Criteria

Attributes and Criteria. Although the framework presented in Buchin et al. (2011) allows for many criteria, we found the basic and intuitive attributes, speed, location, heading, and time, are well suited for describing movement states. For each, we give one (or two) intuitive criteria. Except for the time criterion, these criteria were suggested in Buchin et al. (2011). We will use these criteria in the next section for segmenting migrating geese data. Figure 1 illustrates the criteria location in disk and heading angular range.

Location disk criterion: all points of the segment must lie in a disk of a given radius.
Heading angular range criterion: the angular range of the heading vectors is bounded by a given angle.
Speed minimum/maximum value: all speed values of the segment must be larger/smaller than the given speed value.
Time minimum/maximum duration: the duration of a segment is larger/smaller than the given duration.

Properties. The criteria have (some of) the following properties.

- *monotone*: if a segment fulfills the criterion, then so does each subsegment.
- *linear*: all sample points of a segment need to be checked for the criterion.
- *constant-update*: checking a segment of size one larger can be done in constant time.
- *relative/absolute*: a criterion that depends on relative/absolute values of attributes

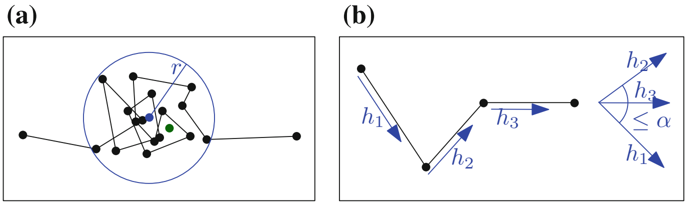


Fig. 1 Illustration of criteria. **a** Location in disk (of radius r). **b** Heading angular range ($\leq \alpha$)

Table 1 Attributes and criteria and the properties they fulfill

Attribute	Criterion	Mono.	Line.	Cons.	Rela.
Location	Disk	Yes	Yes	No	Yes
Heading	Angular Range	Yes	Yes	Yes	Yes
Speed	Minimal value	Yes	Yes	Yes	No
	Maximal value	Yes	Yes	Yes	No
Time	Minimal duration	No	No	Yes	Yes
	Maximal duration	Yes	No	Yes	Yes

Mono. stands for monotone, line. for linear, cons. for constant-update, and rela. for relative

Monotonicity was already discussed in Buchin et al. (2011) and is required for the greedy strategies to be optimal (i.e., result in minimal segmentations). It seems to be a natural requirement, however, there are non-monotone criteria, such as the angle and tube criterion suggested by Turchin (1998) (see below), and criteria on average values.

The properties linear and constant-update affect the runtime of the algorithms. For a linear, constant-update criterion, incremental-search is asymptotically faster than double-and-search, (linear instead of $O(n \log n)$ run time). For a linear criterion with $O(\log n)$ update cost, incremental and double-and-search are asymptotically equally fast.

A relative criterion completely partitions the trajectory, whereas an absolute criterion may give unclassified segments, i.e., pieces not fulfilling the criterion. In a segmentation according to movement states, unclassified segments correspond to segments where the movement states could not be classified.

Table 1 shows which of the criteria listed above are monotone, linear, constant-update, and relative. These properties are all straightforward except for updating heading angular range in constant time, which we describe next.

Heading angular range can be updated in constant time for angles up to 180° using the fact that only one angle between two heading directions needs to be taken into account. That is, one can simply maintain the cone, smaller than 180°, containing all directions, as long as this exists. For larger angles, say $360 - \gamma$, we subdivide the space of angles into $360/\gamma$ ranges of size γ. In each range we maintain the smallest and largest heading direction in constant time and deduce the maximum range from this. The resulting update time remains constant, but the constant increases with decreasing γ.

Further Criteria. There are many further criteria. In Buchin et al. (2011) the authors propose also location diameter, curvature, curviness, sinuosity, and a shape fitting criteria. Turchin (1998) introduced an angle and a tube criterion. Both of the latter criteria use the line from start to end point of a segment as reference line and bound the angle or the location of the points in between with respect to this line. Because they depend on the line from start to end of a segment, these criteria are not monotone.

2.3 Combining Criteria

Describing Movement States. In Buchin et al. (2011) as combination of criteria, a boolean formula in conjunctive normal form or a linear combination was suggested. We propose instead using a boolean formula in disjunctive normal form for describing movement states.

A boolean formula in disjunctive normal form (dnf) describing k different movement states has the form

clause-1 OR clause-2 OR ... OR clause-k

where each clause has the form

criterion-1 AND criterion-2 AND ... AND criterion-l

where the number l of criteria depends on the described movement states. For instance, we may use a dnf formula for describing *flying* or *resting*. If we describe flying as minimum speed and minimum time duration and resting as maximum speed (for a simple example, without parameters), we get in total the formula

(minimum speed AND minimum time duration) OR maximum speed.

Choosing criteria. The number of clauses corresponds to the number of movement states to be used for segmentation (and are thus fixed). For describing a movement state, the more criteria are used, the harder the corresponding clause is to fulfill. Note that a criterion should only be used, if the data contains the corresponding information. For instance, currently GPS data does not contain accurate altitude data, thus, this criterion should not be used in segmentation.

2.4 Algorithms

Basic algorithms. In Buchin et al. (2011) two basic algorithms are suggested: *incremental-search* and *double-and-search* (see Sect. 2.1). Incremental is asymptotically faster for linear, constant-update criteria. For linear, $O(\log n)$-update criteria, the two are asymptotically equally fast. Double-and-search is asymptotically faster for criteria with higher update cost.

Which of the two algorithms is faster for a specific application depends on the trajectory and the set of criteria. If some criteria are linear and constant-update, and some are not, it cannot be said in advance. A naive strategy is to use the algorithm that is better for a majority of the criteria.

Non-monotone criteria. For non-monotone criteria no efficient algorithm is known. For linear, non-monotone criteria, a linear search from the back finds a single longest segment. A linear search from the front (i.e., incremental-search) may result in a non-optimal segmentation in terms of size, i.e., more and shorter segments than necessary may be found.

Continuous Segmentation. For each algorithm, we have the choice between discrete and continuous segmentation, i.e., whether we segment only at sample points or also in between. Segmenting in between sample points in Buchin et al. (2011) is based on the linear motion model, i.e., interpolating linearly in between sampling points. Whether discrete or continuous segmentation is more suitable depends on the application, and in particular on the sampling of the trajectory. If a trajectory is densely sampled, then a discrete segmentation will not differ much from a continuous segmentation. If a trajectory is sparsely sampled, then a discrete and a continuous segmentation may differ considerably. In this case, segmenting in between sample points may be superior. However, for sparse sampling, the linear motion model is not realistic, which applies not only to the location but also to movement parameters (e.g., speed) in between sampling points. Therefore, in our study we chose discrete segmentation.

Extensions. For the two basic algorithms, we developed the following two extensions.

- allowing a min time criterion
- allowing a constant number of outlier

Time Criterion. Minimum time duration is an important, but non-monotone criterion, which is not handled by the original framework. With this criterion it is possible to filter out segments of short duration. Therefore we extend the framework to include it.

Two simple approaches for incorporating it in our framework are:

- Ignore a time criterion until all other criteria of a clause fail. Then test minimum time.
- First find the time threshold and test there.

The first method is simple, but non-optimal. We may discard segments after having tested criteria on them. The second method is faster, if we can find the time threshold, i.e., a segment of the trajectory of the given duration, efficiently (e.g., in regularly sampled data). However, in irregularly sampled data finding time threshold may be time-consuming as well.

Outlier. Outliers, or noise in the data, occur often in todays GPS data. They may influence the analysis, in our case they may cause unwanted cuts in the segmentation, by causing criteria to fail. There are several strategies to deal with these, see also the discussion in Buchin et al. (2011). We propose a simple strategy of ignoring a (small) constant number of points (outliers) per segment in the segmentation. We change our testing strategy, for each criterion separately, to ignore up to a certain amount of points per segment in total, with up to a certain amount of these consecutive. This is useful not only for noise due to GPS error, but also for "relaxing" the criteria describing movement states. Allowing a constant number of outlier points does not affect the monotonicity of criteria, and thus our greedy strategies are still optimal (in the size of the resulting segmentation). A different approach would be to require that only at least a given percentage of a segment fulfills the given criteria. In this case, segmentation is not anymore monotone (extending to the front may allow extending further to the end). Thus, for this we would need other computation approaches, which we leave to future work.

3 Use Case: Migrating Geese

3.1 Geese Data

We implemented our methods (in Java) and tested these on GPS tracks from greater white-fronted geese (Anser albifrons albifrons). The given data of those geese spanned the time of their spring migration (March to June), during which they migrated from the Netherlands to Siberia. The geese where equipped with combined argos/gps microwave tags, and sampled at most every 2 h. See Kruckenberg et al. (2008), van Wijk et al. (2011), and `www.blessgans.de` for a more detailed description of the GPS devices, and `www.blessgans.de` for more information on the data collection.

The segmentation goal for this data set was to segment migration flight and stopovers (including wintering, breeding, and moult). These movement states can be described by spatio-temporal attributes as follows.

flight: little variation in heading, speed at least 20 km/h for at least 5 h
stopover: stay in 30 km radius for at least 48 h

3.2 Geese Criteria

The description of movement states directly translates to the following criteria:

flight: bounded heading angular range AND minimum speed AND min time duration
stopover: location in disk AND minimum time duration

However, because speed values were noisy, we did not use the minimum speed for flight (also not with outliers, see the discussion below). The bounded heading for flight sometimes (artificially) cut pieces of flight into smaller pieces. Due to this, we

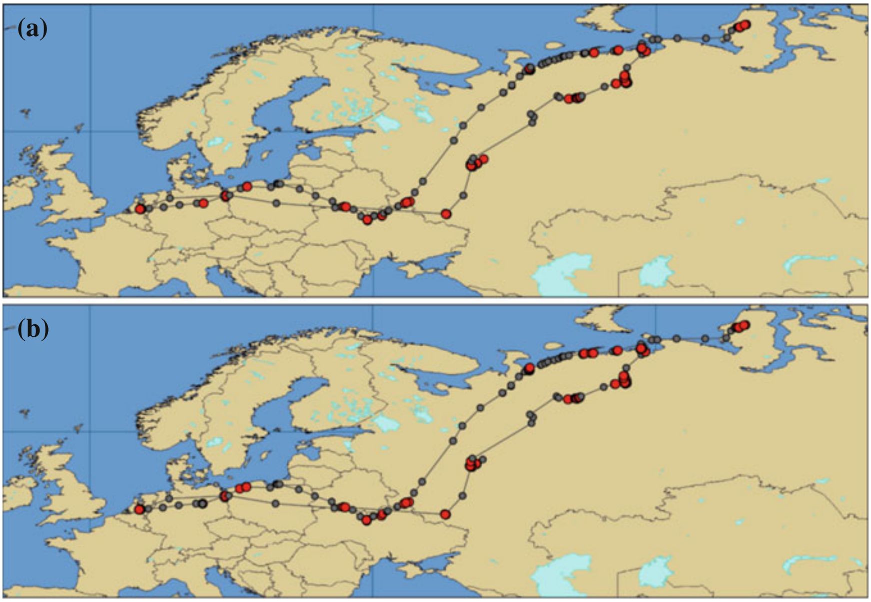


Fig. 2 **a** Manual and **b** computed segmentation of two migrating geese. *Grey* indicates flight, *red* stopover

also omitted the minimum time duration for flight. Otherwise pieces of flight would have been unclassified.

We varied the parameter for heading angular range, i.e., the angle bounding the angular range. We found that 120° gave best results, which coincides with the domain knowledge of this parameter, see the discussion below.

Thus, finally we used the following reduced criteria (with these parameters). Note that these are all relative criteria, thus resulting in a complete segmentation.

flight: bounded heading angular range (120°)
stopover: location in disk (30 km) AND min time duration (48 h)

3.3 Results

From the resulting segmentations (Figs. 2, 3) in comparison to expert classification it becomes clear that the algorithm catches the general pattern of migration flight and stopover in space as well as in time.

Evaluation. The automatic and manual segmentation are very similar on a global scale, that is, the same stopovers are detected. However, there are differences locally. Sometimes short stops are not picked up by the algorithm, but the main stopovers are well detected. Also, the automatic segmentation cuts flight more often, because

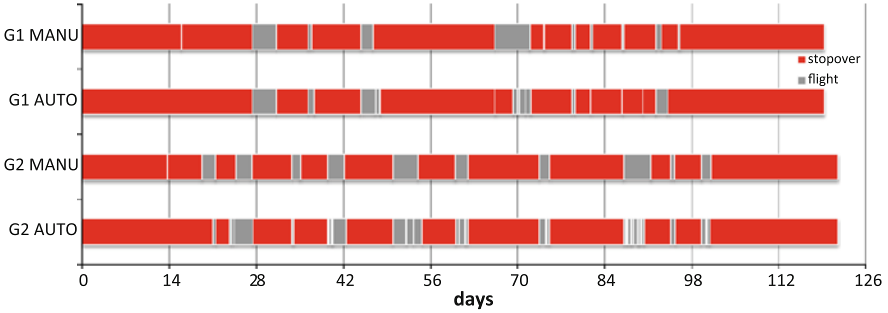


Fig. 3 Comparison of manual and computed segmentation of the tracks of two migrating geese. *Grey* indicates flight, *red* stopover. Day 1 refers to 1 March

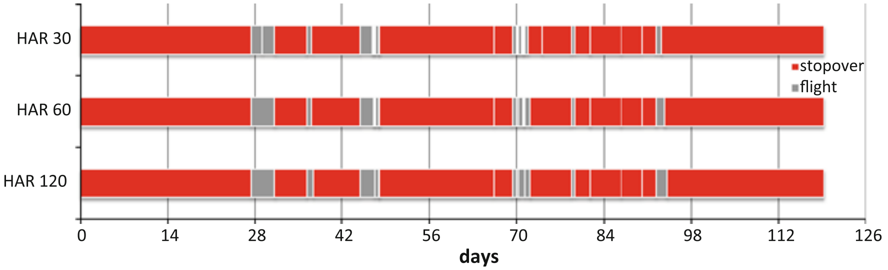


Fig. 4 Varying the parameter for heading angular range (HAR) in the computed segmentation by 120 , 60 %, and 30 %. *Grey* indicates flight, *red* stopover. Day 1 refers to 1 March

of larger variation in heading. The manual segmentation has some longer stopovers, due to taking into account geographical information (e.g. lakes).

Varying parameter. We varied the angle parameter for heading angular range, as demonstrated in Fig. 4. A smaller angle results in a higher segmentation of the pieces of flight, whereas a too large angle results in wrongly classifying pieces of stopover as flight. We choose an angle of 120°, which adds only few (artificial) cutting points, and coincides with the expert knowledge of this parameter, since the geese are known to sometimes change their heading.

We compared the segmentation with and without speed and time criterion for flight, and with and without allowing outliers. However, because of a large amount of noise in the speed values, we decided not to use this criterion, even even with allowing outliers. Since the flight pieces contain small loops, the heading criterion will cut some of the flight pieces. In combination with a minimum time criterion, this results in unclassified pieces (instead of flight). Therefore, we decided also not to use a minimum time, but only heading as criterion for flight.

Coordinate projection. In our current implementation, the location in disk criterion is computed on planar coordinates. The original data, however, is given by latitude and longitude values, which we project (using a Universal Transverse Mercator) to (x, y) coordinates. However, a more precise solution would be to compute the criteria directly based on latitude and longitude values, and instead of Euclidean distance use great-circle or rhumb-line distance. This would be useful, as migration studies use mostly distances based on rhumb lines.

4 Conclusion

The extended segmentation framework for trajectory segmentation based on movement states has proven successful in practice. This is shown by the use case of migrating geese in the previous section. To our knowledge, this is the first automatic approach to segmenting trajectories by spatio-temporal attributes describing movement states.

The success of our approach, however, depends on the possibility to describe movement states with spatio-temporal criteria, and it requires expert knowledge and manual input from the user. This raises the following two questions:

1. How to handle movement states that cannot be described as clearly with the given criteria?
2. How to segment fully automatically, i.e., without input of expert knowledge?

In the first question, the difficulty may come from movement states difficult to describe with spatio-temporal criteria, or because appropriate attributes or criteria are missing. For example, altitude would be useful for flight detection, however, in current GPS altitude data is very imprecise. However, with increasing accuracy of altitude measurements, it would be interesting to include this. Concerning the second question, such a segmentation is not always desirable, however useful for exploratory purposes. We see the strength of our approach in the possibility to manually describe movement states. Other approaches have been developed for (fully) automatic segmentation, e.g., first passage time (Fauchald and Tveraa 2003) and using k-means clustering (van Moorter et al. 2010). In this study, we concentrated on segmentation by spatio-temporal attributes describing movement states of an animal, given the knowledge of a domain expert. If expert knowledge is not available, for instance, methods from machine learning can be employed to complement our approach. In future work, we would like to explore other types of segmentation in movement ecology.

Besides the questions raised above, we plan to include (dynamic) geographic context as a criterion for segmentation. For example, in the case of migrating geese, we would like to use landcover as further criterion for stopovers (grassfields and lakes are important criteria for the geese to stop). Some important context variables are time-dependent, such as grass bloom and snow melt for the migration of geese in this study.

Acknowledgments This research was initiated at the Lorentz Center Workshop on *Analysis and Visualization of Moving Objects*. The authors are grateful to Marc van Kreveld, Emiel van Loon, Judy Shamoun-Baranes, and Bettina Speckmann for helpful discussions. We would like to thank the Vogelschutz-Komitee e.V. (VsK Hamburg), Alterra Wageningen-UR and the Dutch Society of Goosecatchers for the financial and technical support in catching and tracking the geese.

References

Buchin M, Driemel A, van Kreveld M, Sacristán V (2011) Segmenting trajectories: a framework and algorithms using spatiotemporal criteria. J Spatial Inf Sci (3):33–63

Fauchald P, Tveraa T (2003) Using first-passage time in the analysis of area-restricted search and habitat selection. Ecology 84(2):282–288

Kruckenberg H, Müskens G, Ebbinge B (2008) Satellite tracking of greater white-fronted geese anser albifrons during spring migration (2006)—preliminary results, Vogelwelt, pp 338–342

Nathan R, Getz WM, Revilla E, Holyoak M, Kadmon R, Saltz D, Smouse PE (2008) A movement ecology paradigm for unifying organismal movement research. Proc. National Academy of Sciences 105(49):19052–19059. doi:10.1073/pnas.0800375105

Shamoun-Baranes J, van Loon E, Purves R, Speckmann B, Weiskopf D, Camphuysen C (2012) Analysis and visualization of animal movement. Biol Lett 8(1):6–9

Turchin P (1998) Quantitative Analysis of Movement: measuring and modeling population redistribution in plants and animals. Sinauer Associates, Sunderland, MA

van Moorter B, Visscher DR, Jerde CL, Frair JL, Merrill EH (2010) Identifying movement states from location data using cluster analysis. J Wildl Manag 74(3), 588–594 . http://dx.doi.org/10.2193/2009-155

van Wijk RE, Kölzsch A, Kruckenberg H, Ebbinge BS, Müskens GJDM, Nolet BA (2011) Individually tracked geese follow peaks of temperature acceleration during spring migration. Oikos, http://dx.doi.org/10.1111/j.1600-0706.2011.20083.x

The Case for 3D Visualization in DEM Assessment

Michael B. Gousie

Abstract The Digital Elevation Model, or DEM, is a common way to store elevation data. However, errors in various stages of DEM processing mean that the validity of a particular data point is uncertain. In many visualization systems, uncertainty in the data may be highlighted, but it is often difficult for the viewer to discern the exact nature of the problem. DEMView is a prototype DEM display system that incorporates several uncertainty visualizations, including curvature and local differences, while viewing the surface in two or three dimensions. The Profile Cutter and the magnifier are components of the system that allow the user to view a portion of the surface while keeping in the context of the overall area. In addition, the system displays visualizations for several quantitative uncertainty statistics. A detailed case study shows the efficacy of the system, especially the usefulness of viewing in three dimensions.

Keywords DEM · Visualization · Uncertainty · Focus plus context · 3D

1 Introduction

The digital elevation model (DEM), a file format in which elevation values are stored in a regular grid, is commonly used in computer geo-processing. Such data is utilized for many kinds of applications, including emergency route finding, flood plain determination, forest fire management, utility infrastructure, recreational development, and town planning. However, a DEM may be created for a particular geographic location via one of many methods, such as interpolating and/or approximating from contour or sparse data, converting LIDAR or shuttle radar topography mission (SRTM) data, or any number of other photogrammetry techniques. No matter how the DEM

M. B. Gousie (✉)
Wheaton College, Norton, MA 02766, USA
e-mail: mgousie@wheatonma.edu
e-mail: mgousie@wheatoncollege.edu

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_3,

is computed, the accuracy of a particular point may be uncertain. Problems in a DEM can, in turn, lead to dramatic errors in applications that depend on the data. As an added consideration, because of the difficulty of determining the quality of a given DEM and the costs associated with procuring them, many users do not take into account possible errors (Januchowski et al. 2010), thus temporarily avoiding the issue until problems arise in the future.

Many GIS and other software can help users assess the quality of a DEM. However, many of these have a steep learning curve and produce visualizations that are difficult and/or time consuming to evaluate. DEMView is a prototype system built solely for the purpose of viewing DEMs and assessing errors, and by its not having multiple layers of menus, is easier to navigate. It offers several quantitative and qualitative assessment tools, including visualizations in two or three dimensions, giving the user flexibility by offering various views that may help shed light on any potential problems in a DEM. A "profile cutter" and magnifier are two tools that allow the user to see small scale details in 2D within the context of a 3D visualization. A detailed case study is presented that highlights the major components of the system.

2 Related Work

The problem of assessing error and/or uncertainty in a DEM can be divided into two parts: (1) quantifying the error and (2) producing a visualization for assessed errors. Various approaches to ascertaining the extent of DEM error have been proposed (Fisher and Tate 2006), many of which are outlined below.

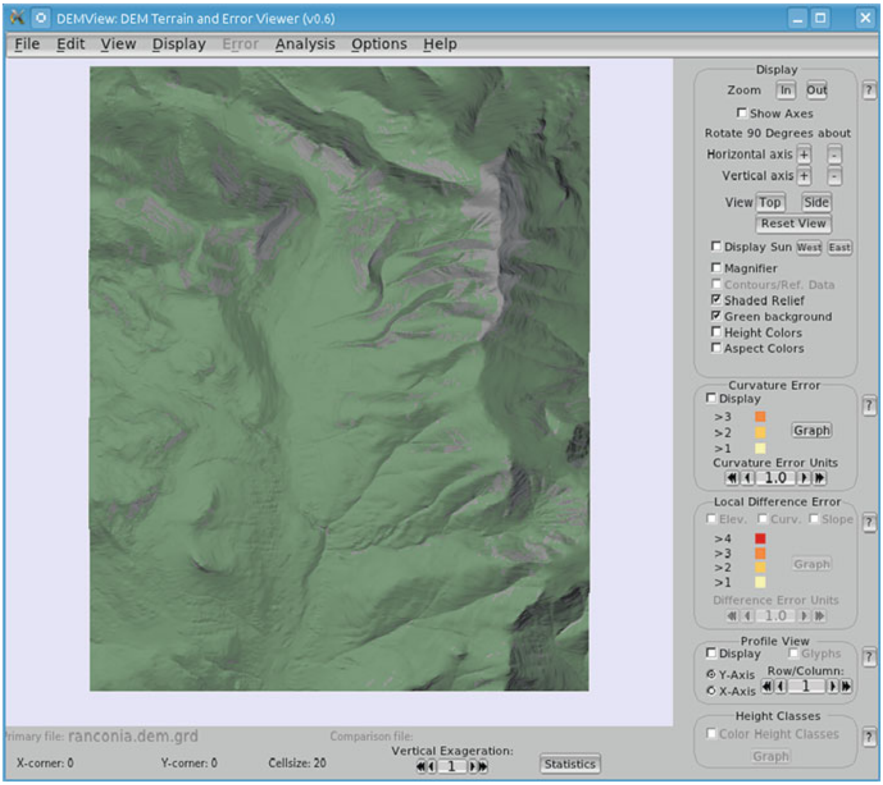
A standard uncertainty measure is the root mean square error (RMSE), which compares a DEM height point with a corresponding elevation from an accurate source (Alam and Luding 1988). Although it gives only a global measure of the validity of a DEM, recently Wise (2011) found that RMSE of elevation is a good predictor of RMSE in gradient and aspect. Carrara et al. (1997) use several analysis techniques, including determining if DEM heights fall between contour elevations. One way to test this is to create profile plots with the contour elevations highlighted (Gousie and Franklin 2005), while another method is to use elevation histograms to show if there is a linear fit between contours (Carrara et al. 1997; Reichenbach et al. 1993). One can also compute the smoothness of a DEM by computing the total squared curvature (Briggs 1974) or, similarly, finding local curvature. Fisher (1998) computed several statistics after comparing a DEM with established spot heights and computes a probable viewshed. Errors, based on grid bias, can be found by comparing drainage networks extracted by multiple rotations of the DEM (Charleux-Demargne and Puech 2000). Rigorous statistical models have been proposed as well (Carlisle 2005). Many of the above methods require the user to interpret the resulting error data. A visualization of the error gives the viewer immediate feedback to potential problems. Wood and Fisher (1993) were early proponents of such visualizations; they compared several interpolated DEMs by displaying visualizations of aspect, Laplacian filtering that highlights sudden changes in elevation, RMSE, and shaded relief. While these give the viewer good insight not only to what the problems are but exactly where they lie, the visualizations were rendered in only two dimensions. Much work has been done in uncertainty visualization, such as using glyphs, translating/rotating surface patches to highlight potential error, altering lighting parameters, and so forth (Johnson and Sanderson 2003; Pang et al. 1996). Kao et al. show ways to visualize 2D probability distributions from geo-data sets (Kao et al. 2001; Luo et al. 2003). MacEachren et al. give a comprehensive overview of the state of visualizing uncertainty in geospatial domains (MacEachren et al. 2005).

There are many GIS that have good 3D visualization capability and at least some uncertainty visualization features, of which the following is a sampling. Textures are shown to be useful for terrain visualization (Döllner et al. 2000). Terrafly (Rishe et al. 2004) displays satellite imagery and other data in various resolutions. GeoZui3D (Ware et al. 2001) is a 3D marine GIS that supports multiple linked views; that is, the user can view the overall area and a smaller portion at much greater resolution. A GIS that integrates 2D and 3D views of the same data is described in Brooks and Whalley (2005). A system that incorporates some error capabilities is LandSerf (Raper et al. 2002; Wood 1996), including shaded relief, curvature visualization, peak classification, and others. LandSerf is also very useful in generating contours and reading/writing many file formats. Another tool dedicated to displaying geographic areas and some errors using orthoimages is described in Wiggenhagen (2000). A thorough statistical comparison between a DEM computed from contours and new LIDAR shows that DEM error is indeed present and comes from several sources (Oksanen and Sarjakoski 2006). This work also shows the usefulness of visualizations in detecting and evaluating errors. VisTRE (Healey and Snoeyink 2006) is a system designed expressly for visualizing terrain errors. The work is guided by psychophysical studies to maximize the effectiveness of the visualizations while limiting perceptual biases.

3 DEMView Assessment and Visualization Tools

DEMView is a prototype system for DEM uncertainty visualization in two and three dimensions, written in C++ with the OpenGL Application Programming Interface (API) for the graphics rendering, the OpenGL Utility Toolkit (GLUT) for the window system and stenciling (see below), and FLTK (Fast Light Toolkit) (Easy software 2007) for the graphical user interface (GUI). Figure 1 shows the system displaying a 1000 × 1200 10-m DEM taken from the 7.5' USGS National Elevation Dataset (NED) covering Franconia, NH. Elevations are in feet. The program reads data files in standard ArcInfo ASCII grid format. The default visualization shows the surface in green shaded-relief, with gray in areas above user-defined tree line elevation as well as in steep-slope terrain. Turning the green background off yields an all-gray shaded relief map.

A feature of the system is that the GUI is designed specifically for visualizing uncertainty in DEMs, somewhat following the model used by LandSerf except that



Display
Zoom In Out ?
□ Show Axes
Rotate 90 Degrees about
Horizontal axis + -
Vertical axis + -
View Top Side
Reset View
□ Display Sun West East
□ Magnifier
□ Contours/Ref. Data
☑ Shaded Relief
☑ Green background
□ Height Colors
□ Aspect Colors

Curvature Error ?
□ Display
>3 🟧
>2 🟨 Graph
>1 🟨
Curvature Error Units 1.0

Local Difference Error ?
□ Elev. □ Curv. □ Slope
>4 🟥
>3 🟧
>2 🟨 Graph
>1 🟨
Difference Error Units 1.0

Profile View ?
□ Display □ Glyphs
● Y-Axis Row/Column: 1
○ X-Axis

Height Classes ?
□ Color Height Classes
Graph

rimary file: ranconia.dem.grd
Comparison file:
X-corner: 0 Y-corner: 0 Cellsize: 20
Vertical Exageration: 1
Statistics

Fig. 1 Default view of franconia DEM on DEMview. Note GUI panel on *right* showing all options

the results of all operations can be viewed on the surface in three dimensions. All available features are displayed on the right panel at all times (unless hidden by user); they are all available through menus as well. Options that are grayed-out require a second comparison DEM (see below) for activation. Rotation of the 3D surface is accomplished through the left mouse button and translation through the right button. Zooming can be done using the panel buttons or the scroll button on a mouse. Other panel buttons provide common rotations/translations with one-click functionality. Contour, sparse, or full DEM data can be used to compare with the subject DEM, with various ways to display both data sets simultaneously as described below. In all cases, no special scripts or multiple levels of menus are required.

3.1 Curvature and Local Difference Error Visualization

The overall smoothness of a DEM can be computed by finding the total squared curvature, C_{sq} (Briggs 1974):

$$C_{sq} = \sum \sum (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j})^2 \tag{1}$$

The total squared curvature may be biased if there are large problem areas in a DEM. To mitigate this, an indication of local smoothness can be found by averaging the local, or absolute, curvature which is found at a point i, j:

$$C_{abs} = |(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j})| \tag{2}$$

The value of C_{abs} is the curvature at a specific point. Severe curvature may indicate an error in the DEM; patterns in curvature may indicate systematic errors due to interpolation. Such curvature error can be displayed in DEMView, where the threshold is chosen by the viewer. The curvature is displayed via different hues, where the green surface indicates no error (little curvature) and progressing through yellow to orange for the highest error (extreme curvature). The colors were chosen in accordance with other visualization systems (Healey and Snoeyink 2006) and color perception studies (CIE 1978). The user may choose to have these errors categorized into clear levels or displayed via a change in hue proportional to the error. The GUI labels change dynamically to indicate the current level of error relative to the displayed colors.

To visualize discrepancies between source data and DEM, each source height point is compared to the corresponding elevation in the DEM to find the local difference error d at point i, j:

$$d_{i,j} = |u_{i,j} - v_{i,j}| \tag{3}$$

where v is the elevation in the comparison DEM. Following Carrara et al. (1997), d should not be greater than 5 % of the contour interval. Thus, similar to the mechanism described for curvature error, colors are assigned to elevations that have d greater than 5, 10 %, and so forth. If the comparison data is sparse data or another DEM, the user can indicate an appropriate level of error.

3.2 Height Class Frequency Visualization

If the source data is contours, then the DEM values within an area bounded by a contour pair should vary almost linearly, indicating an absence of artifacts such as terracing. DEM elevations are grouped into integer intervals between two contours and then reclassified into relative elevations (Carrara et al. 1997). For example, if

1200–1220 represents a contour pair, then the relative elevations, or height classes, would be 0, 1, 2, …, 19 corresponding to the elevations of 1200, 1201, 1202, …, 1219. The height classes are computed and the surface is displayed in green with the absolute frequency of the relative heights shown in graduated color from green to orange. The brighter the orange, the higher the absolute frequency of that height class, indicating that the slope is not linear between successive contours. The actual absolute frequencies are displayed as well for graphing purposes. It must be noted that the absolute frequency is a global measure that is applied to individual points, and thus the visualization is only a guide as to where errors may be. In other words, all points with the same color indicate they are all in the same height class. Ideally, there should be no orange in the surface at all.

3.3 Quantitative Statistics

The user can also opt to have DEMView display various statistics. These include total squared curvature, maximum curvature, and the count for each of the curvature error levels (which can also be shown as a graph), along with other standard summary statistics. If the DEM is being compared to another DEM or sparse data, comparison statistics are computed as well. These include the counts for difference error levels and the RMSE. Additional statistics will be included in future versions of the system.

4 Focus Plus Context

DEMView supports two tools that allow the user to focus on a small portion of a DEM while keeping the surface in context. The Profile Cutter and the magnifier can be used in conjunction with any other visualizations described above.

4.1 The Profile Cutter

While many systems offer visualizations that enable the viewer to see errors in general, it is often difficult to zoom in on a small area to ascertain minute differences between a DEM and comparison data. The Profile Cutter is a semi-transparent planar rectangle that is orthogonal to the surface. The cutter enables the viewer to make a vertical “slice” through the DEM to better see the profile at any x or y position. The position can be changed dynamically through buttons on the GUI, including moving the profile incrementally. Alpha blending makes the cutter semi-transparent, thus showing the profile within the context of the remaining DEM in the background. We also implemented what might be called “full context;” that is, showing the DEM

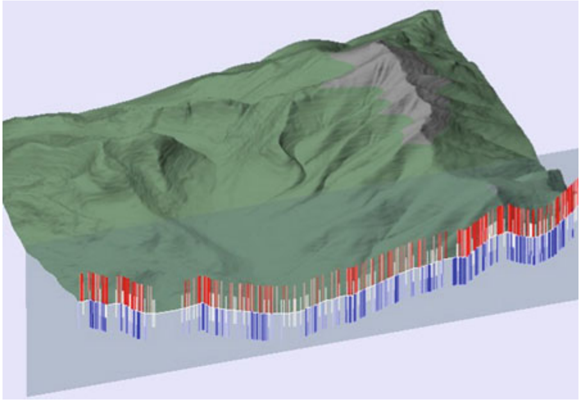


Fig. 2 The profile cutter slicing through the franconia DEM

portion in front of the Profile Cutter as a semi-transparent surface, but this made the visualization too cluttered.

The power of the Profile Cutter is more apparent when the primary DEM is being compared to another data set, be it sparse or another full DEM. The profile that is shown in white is always that of the primary DEM. In the comparison data set, if there exists a valid elevation at an x, y position, then a glyph can be displayed in the profile. The glyph is a vertical line segment of constant length that has the following properties:

- If the primary and secondary elevations match within a user-specified threshold, then the glyph is rendered in white vertically centered at the profile.
- If the elevation in the primary DEM is below the elevation in the secondary, then the glyph is rendered in a red hue proportional to the difference of the two elevations, where almost white indicates a slight difference and bright red indicates a large difference. In addition, the bottom endpoint of the line segment is at the elevation contained in the secondary data set.
- If the elevation in the primary DEM is above the elevation in the secondary, then the glyph is rendered in shades of blue, with dark blue indicating a large difference. The endpoint at the top of the line segment is at the elevation contained in the secondary data set.

Figure 2 shows the Profile Cutter slicing through the Franconia DEM compared to contour data. The glyphs show how well the contour elevations match the DEM. More examples of the Profile Cutter are shown in the case study.

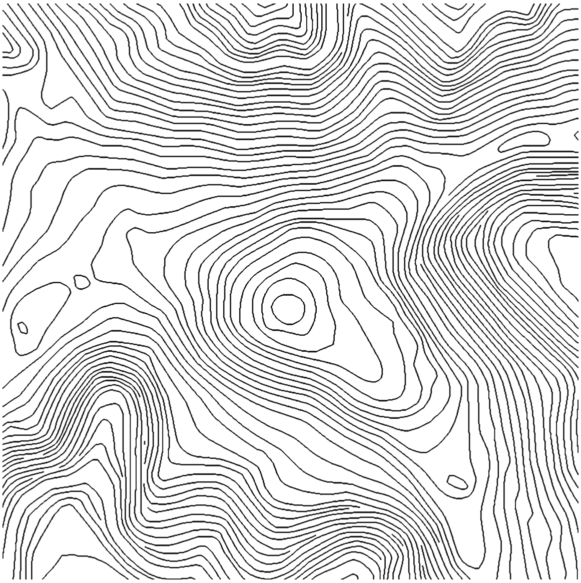


Fig. 3 Contours of Mt. Washington; Tuckerman Ravine is shown in the SW corner

4.2 The Magnifier

The newest tool in the system, and still in its early stages, is the magnifier. This tool enables the viewer to zoom in on just one portion of the DEM and any enabled visualizations being viewed in three dimensions, thereby keeping this zoomed area within the context of the overall terrain. This may be useful in getting a closer look at a possible problem area without losing one's place in the DEM. This idea comes from Magic Lenses (Bier et al. 1993), which could not only magnify but could also be used as an effective interface tool. A 3D version was implemented soon afterward (Viega et al. 1996). Looser et al. extended the lenses for augmented reality interfaces (Looser et al. 2004). Detail lenses, a similar idea for zooming in on areas for route visualization is described in Karnick et al. (2010), but its use is limited to 2D. Studies have also shown that semantic lensing and/or focus plus context are beneficial for tasks similar to what is being presented here (Baudisch et al. 2002; Kalghatgi et al. 2006). The magnifier is activated by pressing the appropriate button; clicking on the middle button or scroll wheel positions and/or drags the magnifier. The area within the magnifier is enlarged by a factor of two. The amount of zoom will be user-defined in future versions. The underlying visualization continues to be in 3D, and all transformation functions are available, allowing the user to dynamically change the viewing position with the magnifier on.

The implementation of the magnifier uses a stencil buffer. The user positions the cursor on the DEM; from this position a rectangular window is defined. The entire surface is zoomed but is then clipped to the window. Thus, only the zoomed portion is rendered (along with the original DEM behind), offering dynamic performance. A circular magnifier, in keeping with many people's notion of a hand-held magnifier, has been studied, but its implementation may be too inefficient for reasonable performance.

The use of the magnifier is demonstrated in the Case Study, below.

5 Case Study

Consider Fig. 3, an 800 × 800 DEM with 1 m resolution constructed from a USGS DLG of Mt. Washington, NH, with 20 m contour intervals. This mountain has the distinction of being the highest peak in the Northeast United States as well as having a dangerous reputation because of severe weather and avalanche danger. In fact, 13 people have died on the mountain since 1956 due to avalanches (Mount Washington 2012). Several deaths have occurred in the Tuckerman Ravine area, a popular spring skiing venue that often exhibits dangerous snow conditions. One may wish to investigate the terrain in that area to determine the causes of those avalanches.

Now further suppose that the aforementioned contours are the sole data available. In order to investigate the area more fully, two DEMs were produced by interpolating the contours using two methods: TOPOGRID (Hutchinson and Gallant 1999; Hutchinson 1988), a well-known and reliable method available in Arc/Info, and an algorithm whereby intermediate contours (INTERCON) are first generated before interpolating (Gousie and Franklin 2005). Figure 4 shows the TOPOGRID surface while Fig. 5 shows the INTERCON DEM. In both cases, the DEMView displays gray above the 4500 foot treeline and green below; note that this feature can be toggled. Looking at the two figures, there are clearly differences between the two DEMs. Which one should be used for further study? The next sections describe the functionality of DEMView and show the effectiveness of the three-dimensional viewing.

5.1 Preliminary Statistics

In order to do a preliminary assessment, the TOPOGRID and INTERCON DEMs were loaded into DEMView along with the original contours. Table 1 shows the relevant statistics. The RMSE shows the fit of each DEM with the original contours; INTERCON is clearly better, but not an exact interpolation of the data. The total squared curvature is also much less than TOPOGRID's, corroborating the visual sense of smoothness. The curvature class counts reflect the number of points that have a local curvature of over three, over two, and over one. These counts reflect the overall curvature, or roughness, that can be seen in Figs. 4 and 5. The elevation

Table 1 Statistical comparison of TOPOGRID and INTERCON DEMs

Statistic	TOPOGRID	INTERCON
RMSE	3.39	1.19
Total C_{sq}	134142.30	18582.44
Max C_{abs}	30.24	2.73
Average C_{abs}	0.21	0.03
Curvature class counts		
>3.0	1732	0
>2.0	2368	117
>1.0	15742	2104
Elevation difference class counts		
>4.0	3669	550
>3.0	1127	577
>2.0	1721	2252
>1.0	3578	11007

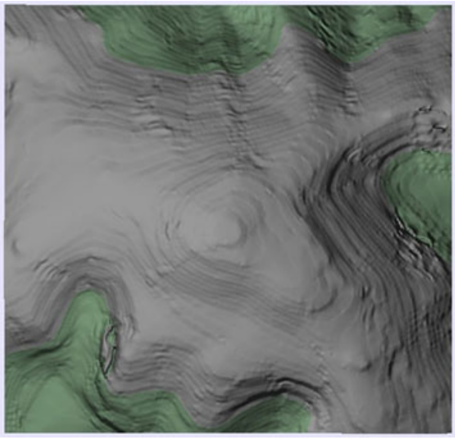


Fig. 4 TOPOGRID DEM of Mt. Washington

difference class counts show the number of points that deviate by more than four meters, more than three meters, etc. Interestingly, the TOPOGRID surface has fewer points in total that deviate from the original contour data, but for those points that do, there are a significant number whose local differences are much worse than INTERCON's. But *where* are the differences? If we wish to study an avalanche area, it is crucial to know where the DEM problems may lie. Figures 6 and 7 show DEMView's curvature visualization turned on. The colors change from yellow (low curvature) to bright orange (high curvature). Clearly, there is much more local curvature in the TOPOGRID DEM. Similarly, Figs. 8 and 9 show the TOPOGRID and INTERCON DEMs with the local difference visualization turned on. The results are much the same; that is, the TOPOGRID surface shows more points with a high elevation difference compared to the original contour data. While these tools show that there is

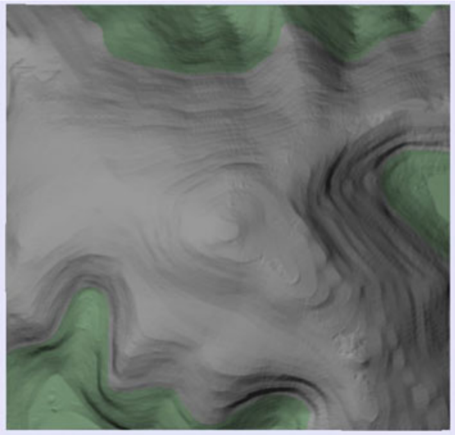


Fig. 5 INTERCON DEM of Mt. Washington

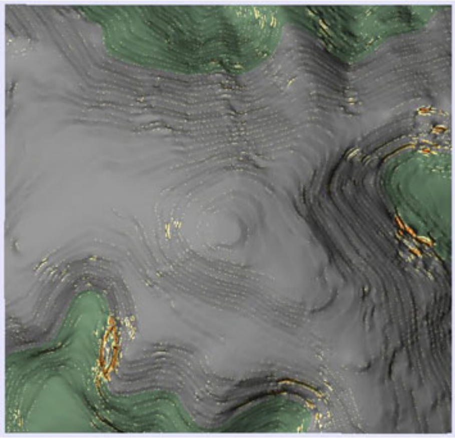


Fig. 6 TOPOGRID showing curvature. Colors range from *yellow* (low curvature) to *orange* (high curvature). Note problematic section in SW corner

indeed some anomaly in the TOPOGRID DEM especially, it is not visually clear what the problem may be.

5.2 3D Visualization Tools

Although much statistical and visualization analysis can be done with DEMs shown only in two dimensions, it may be beneficial to give the user the option of viewing in three dimensions. This extra functionality may shed more light on a particular problem area of a DEM. For example, Fig. 10 shows a zoomed and rotated view of the Tuckerman Ravine area of Fig. 8. This now clearly demonstrates the strange "bulge"

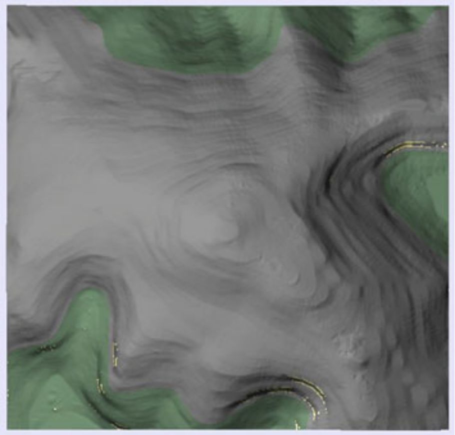


Fig. 7 INTERCON showing far less curvature than TOPOGRID

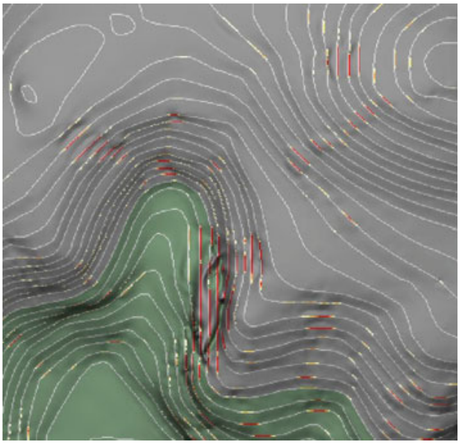


Fig. 8 DEMview displaying elevation differences between zoomed SW corner of the TOPOGRID DEM and underlying contours (shown in gray). Colors range from *yellow* (small difference) to *red* (large difference)

that was a result of the TOPOGRID interpolation; furthermore, the interpolation artifacts along the contours themselves are more easily visible.

Another functional aspect of DEMView is the ability to layer two DEMs on top of one another and then compute and display the local elevation differences and local curvature differences. Figure 11 shows elevation differences between the TOPOGRID and INTERCON DEMs. The obvious red section shows major differences between the elevations of the two DEMs in that area (refer back to Fig. 9).

Another use of three-dimensional visualization is shown in Figs. 12 and 13. The former shows the Profile Cutter slicing through the problem area. Note that (a) the surface behind the cutter provides context of where the profile is being cut; this is much better than many systems that allow for profiles but not in context (LandSerf for example), and (b) both the TOPOGRID (in white) and the INTERCON profiles (in blue and red) are shown simultaneously. Thus the viewer can ascertain that the

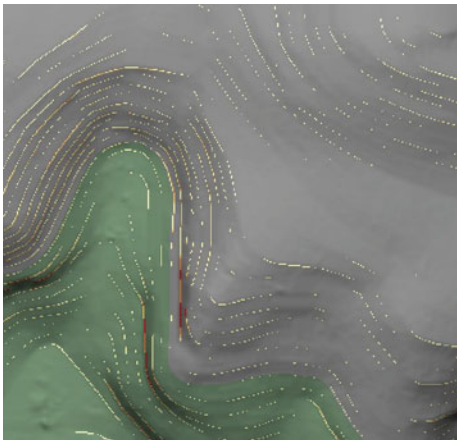


Fig. 9 INTERCON showing less drastic elevation differences but in higher quantity

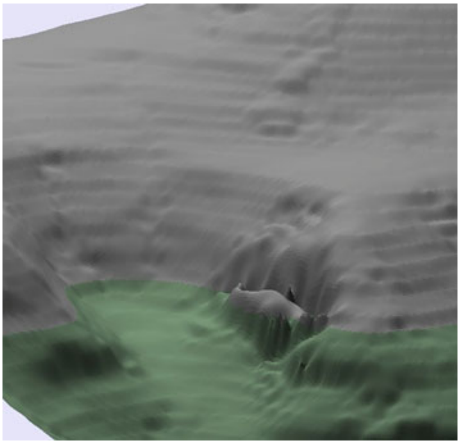


Fig. 10 Rotated and zoomed view of Tuckerman Ravine

INTERCON surface has a much more natural curve in that area than the TOPOGRID DEM. The latter figure shows the same profile but with the glyphs turned on. This example shows all of the possibilities: the white glyphs represent agreement between the two DEMs, the shaded red glyphs (redder = larger difference) indicate that the primary file (TOPOGRID in this case) has lower elevations than the comparison DEM (INTERCON), and shaded blue glyphs indicate higher elevations in the primary DEM. Finally, Fig. 14 shows the magnifier highlighting the problem area on the Mt. Washington DEM. The magnifier affords close-up views similar to previous figures in the context of the entire surface. The magnifier itself may be moved dynamically and works in conjunction with the assessment visualizations.

In all of these views, it would seem that the INTERCON DEM is better suited for further study of the Tuckerman Ravine area, as the surface exhibits fewer anomalies.

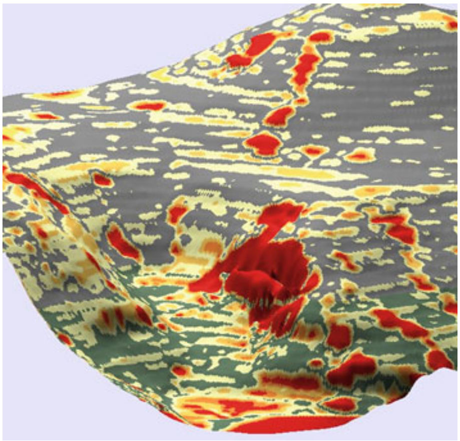


Fig. 11 Elevations differences between TOPOGRID and INTERCON

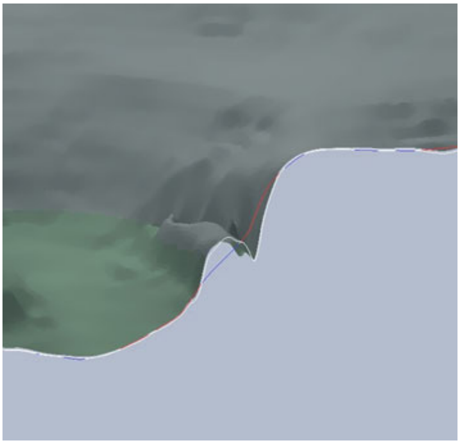


Fig. 12 Profile cutter showing profiles of TOPOGRID (*white*) and INTERCON (DEMs)

6 Conclusion and Future Work

Here we have presented DEMView, a DEM and error visualization system. The curvature and local difference visualizations aid the user in finding areas of uncertainty in a DEM. The Profile Cutter can help the user more clearly see the anomalous regions, as well as compare one DEM to another in a very specific area, all while keeping in context of the entire surface. The magnifier further aids the visualization. In using the tools, especially in conjunction with 3D viewing that afford additional information not seen in 2D, users can better decide how well a DEM suits their needs. Furthermore, all of DEMView's functionality are easily accessible through the right panel, obviating the need to search through menus, etc.

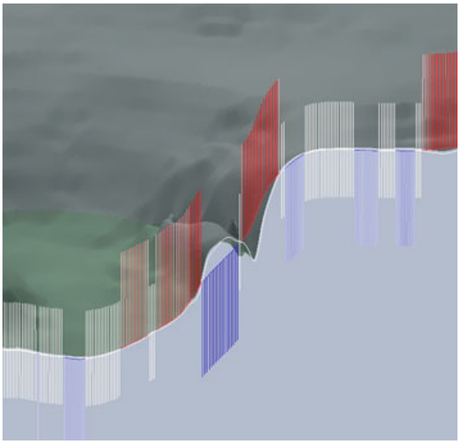


Fig. 13 Same comparison as in Fig. 12, but with glyphs turned on. The brightness of the colors indicate the magnitude of the elevation difference

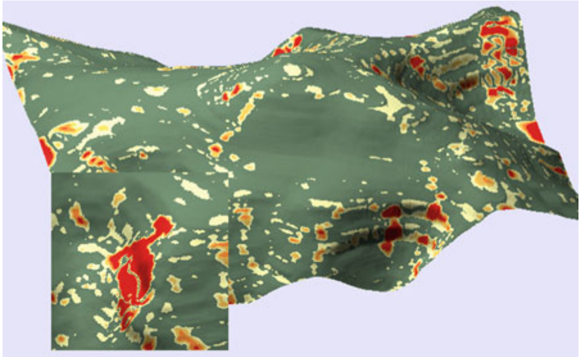


Fig. 14 Comparison of two DEMs of Mt. Washington with magnifier turned on; *red* indicates areas with a poor match between the two

In the future, additional visualizations of spatial statistics will be investigated. The magnifier is of special interest; current and new uncertainty visualizations could be rendered through the magnifier window, similar to Magic Lenses (Bier et al. 1993), allowing the user to remain in context at all times while affording dynamic "browsing" with the mouse over the surface. Another idea is to have the system find a cluster of local error and automatically focus on such an area with the magnifier tool. Finally, robust user studies would be useful to quantitatively determine the system's ease of use.

References

Baudisch P, Good N, Bellotti V, Schraedley P (2002) Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In: Proceedings of the SIGCHI conference on human factors computing systems: Changing our world, changing ourselves (CHI '02). ACM, pp 259–266

Bier EA, Stone MC, Pier K, Buxton W, DeRose TD (1993) Toolglass and magic lenses: the see-through interface. In: Proceedings of the 20th annual conference on computer graphics and interactive techniques (SIGGRAPH '93), ACM, pp 73–80

Briggs I (1974) Machine contouring using minimum curvature. Geophysics 39(1):39–48

Brooks S, Whalley JL (2005) A 2d/3d hybrid geographical information system. Proceedings of ACM graphite, Dunedin, New Zealand, In, pp 323–330

Carlisle BH (2005) Modelling the spatial distribution of DEM error. Trans GIS 9(4):521–540

Carrara A, Bitelli G, Carla' R (1997) Comparison of techniques for generating digital terrain models from contour lines. Int J Geog Info Sci 11(5):451–473

Charleux-Demargne J, Puech C (2000) Quality assessment for drainage networks and watershed boundaries extraction from a digital elevation model (dem). In: 8th ACM symposium on GIS, Washington, D.C. pp 89–94

CIE: CIE publication No. 15, Supplement Number 2 (E-1.3.1, 1971) (1978) Official recommendations on uniform color spaces, color-difference equations, and metric color terms. Commission Internationale de L'Èclairge

Döllner J, Baumann K, Hinrichs K (2000) Texturing techniques for terrain visualization. In: Proceedings of the 11th IEEE visualization conference (VIS 2000), pp 227–234

Easy software products: FLTK: Fast light toolkit. http://www.fltk.org/index.php, http://www.fltk.org/index.php. Accessed 13 March 2007

Fisher P (1998) Improved modeling of elevation error with geostatistics. GeoInformatica 2(3): 215–233

Fisher PF, Tate NJ (2006) Causes and consequences of error in digital elevation models. Prog Phys Geogr 30(4):467–489

Gousie MB, Franklin WR (2005) Augmenting grid-based contours to improve thin plate DEM generation. Photogram Eng Remote Sens 71(1):69–79

Healey CG, Snoeyink J (2006) Vistre: a visualization tool to evaluate errors in terrain representation. In: Proceedings, 3rd international symposium on 3D data processing, visualization and transmission (3DPVT 2006), Chapel Hill, North Carolina

Hutchinson MF, Gallant JC (1999) Representation of terrain. In: Longley, Goodchild, Maguire, Rhind (eds) Geographical information systems: principles and technical issues, vol 1, 2 edn. Wiley, New York, pp 105–124

Hutchinson MF (1988) Calculation of hydrologically sound digital elevation models. Proceedings of the third international symposium on spatial data handling, Int Geog Union, Columbus, Ohio, In, pp 117–133

Januchowski S, Pressey R, VanDerWal J, Edwards A (2010) Chracterizing errors in digital elevation models and estimating the financial costs of accuracy. Int J Geog Info Sci 24(9):1327–1347

Johnson CR, Sanderson AR (2003) A next step: visualizing errors and uncertainty. IEEE Comput Graphics Appl 23(5):6–10

Kalghatgi N, Burgman A, Darling E, Newbern C, Recktenwald K, Chin S, Kong H (2006) Geospatial intelligence analysis via semantic lensing. In: CHI '06 extended abstracts on human factors in computing systems (CHI EA '06), ACM, pp 935–940

Kao D, Dungan JL, Pang A (2001) Visualizing 2d probability distributions from eos satellite image-derived data sets: a case study. In: Proceedings of the conference on visualization '01 (VIS '01), IEEE, pp 457–461

Karnick P, Cline D, Jeschke S, Razdan A (2010) Route visualization using detail lenses. IEEE Trans visual comput graphics 16(2):235–247

Looser J, Billinghurst M, Cockburn A (2004) Through the looking glass: the use of lenses as an interface tool for augmented reality interfaces. In: Proceedings of the 2nd international conference on computer graphics and interactive techniques in australasia and South East Asia (GRAPHITE '04), ACM, pp 204–211

Luo A, Kao D, Pang A (2003) Visualizing spatial distribution data sets. In: Proceedings of the symposium on data visualization 2003 (VISSYM '03), Eurographics Association, pp 29–38

MacEachren AM, Robinson A, Hopper S, Gardner S, Murray R, Gahegan M, Hetzler E (2005) Visualizing geospatial information uncertainty: what we know and what we need to know. Cartogr Geog Inf Sci 32(3):139–160

Mount Washington Observatory: Surviving mount washington (6/2012), http://www.mountwashington.org/about/visitor/surviving.php. Accessed 6 2012

Oksanen J, Sarjakoski T (2006) Uncovering the statistical and spatial characteristics of fine toposcale DEM error. Int J Geog Inf Sci 20(4):345–369

Pang AT, Wittenbrink CM, Lodha SK (1996) Approaches to uncertainty visualization. Visual Comput 13(8):370–390

Raper J, Dykes J, Wood J, Mountain D, Krause A, Rhind D (2002) A framework for evaluating geographical information. J Inf Sci 28(1):51–62

Reichenbach P, Pike RJ, Acevedo W, Mark RK (1993) A new landform map of italy in computer-shaded relief. Bollettino Geodesia a Scienze Affini 52:22–44

Rinehart RE (1988) Coleman EJ digital elevation models produced from digital line graphs. In: Proceedings of the ACSM-ASPRS annual convention. American congress on surveying and mapping, American Society for Photogrammetry and Remote Sensing 2:291–299

Rishe N, Sun Y, Chekmasov M, Andriy S, Graham S (2004) System architecture for 3d terrafly online gis. In: Proceedings of the IEEE sixth international symposium on multimedia software engineering (MSE2004), www.terrafly.com. pp 273–276

Viega J, Conway MJ, Williams G, Pausch R (1996) 3d magic lenses. In: Proceedings of the 9th annual ACM symposium on user interface software and technology(UIST '96), ACM, pp 51–58

Ware C, Plumlee M, Arsenault R, Mayer LA, Smith S (2001) Geozui3d: data fusion for interpreting oceanographic data. In: Proceedings of the MTS/IEEE conference and exhibition (OCEANS 2001), vol 3. pp 1960–1964

Wiggenhagen M (2000) Development of real-time visualization tools for the quality control of digital terrain models and orthoimages. In: International archives of photogrammetry and remote sensing (IAPRS), vol 33. Amsterdam, pp 987–993

Wise S (2011) Cross-validation as a means of investigating DEM interpolation error. Comput Geosci 37:978–991

Wood JD (1996) The geomorphological characterisation of digital elevation models. Ph.D thesis, University of Leicester, UK

Wood JD, Fisher PF (1993) Assessing interpolation accuracy in elevation models. IEEE Comput Graphics Appl 13(2):48–56

A Hybrid Shortest Path Algorithm for Intra-Regional Queries on Hierarchical Networks

Gutemberg Guerra-Filho and Hanan Samet

Abstract A hierarchical approach to the single-pair shortest path problem subdivides a network with n vertices into r regions with the same number m of vertices ($n = rm$) and iteratively creates higher levels of a hierarchical network by merging a constant number c of adjacent regions. In a hierarchical approach, shortest paths are computed at higher levels and expanded towards lower levels through intra-regional queries. We introduce a hybrid shortest path algorithm to perform intra-regional queries. This strategy uses a subsequence of pre-processed vertices that belong to the shortest path while actually computing the whole shortest path. At the lowest level, the hybrid algorithm requires $O(m)$ time and space assuming a uniform distribution of vertices. For higher levels, the path view approach takes $O(1)$ time and requires $O(c^k m)$ space.

1 Introduction

Information systems that assist drivers in planning a travel are required to improve safety and efficiency of automobile travel. These systems use real-time traffic information gathered by traffic control and surveillance centers like traffic congestion and roadwork. They aid travelers in finding the optimal path to their destinations considering distance, time and other criteria. This helps to eliminate unnecessary travel time reducing accidents and pollution.

This work was supported in part by the National Science Foundation under Grants IIS-10-18475, IIS-09-48548, IIS-08-12377, and CCF-08-30618.

G. Guerra-Filho (✉)
Department of Computer Science and Engineering, University of Texas, Arlington, TX, USA
e-mail: guerra@cse.uta.edu

H. Samet
Department of Computer Science, University of Maryland, College Park, MD, USA

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_4,

A *Moving Object Database* (MOD) is a special form of a spatial database that represents information about moving objects, their location in space, and their proximity to other entities or objects. In spatial database applications, the embedding space consists in a geographical network with a distance metric based on shortest paths. Hence, shortest path finding is a basic operation in MODs. This operation is used as a subroutine by many other proximity queries, including nearest neighbors (Kolahdouzan and Shahabi 2004). In particular, finding nearest neighbors in a spatial network presumes that the shortest path to the neighbors have been computed already. The efficiency issues related to this operation are critical to MODs due to the dynamic and real-time characteristics of such databases. A large number of queries in a huge network may prevent the system from meeting the real-time requirement when a non-efficient approach is used.

A *path view* or *transitive closure* contains the information required to retrieve a shortest path corresponding to each pair of vertices in the network. This strategy pre-computes all shortest paths in the network. Once the path view is created, any path query is performed with a lookup in the path view and reporting the sequence of vertices that represent the path. A path query takes $O(n)$ time using a path view, since the number of vertices in the path may be linear in the worst case. The path view strategy requires $O(n^2)$ time for pre-processing (Frederickson 1987; Henzinger et al. 1997) and $O(n^2)$ space. The quadratic space is achieved when a predecessor matrix represents the path view. Methods that pre-compute and store the shortest paths between every pair of vertices in a graph assume that the real-time computation of the shortest paths for large networks may not be feasible. However, the focus of their work is the compact encoding of the $O(n^2)$ shortest paths and efficient retrieval. However, a major drawback of path view approaches is that large networks may need an unacceptable amount of time and space in order to satisfy the real-time constraint.

A hierarchical approach for shortest path finding is one possible way to satisfy both time and space efficiency requirements for moving object databases based on large geographical networks. A *hierarchical approach* subdivides a single network into r smaller regions. Each such region has the same number m of vertices and belongs to the lowest level of the hierarchy. The same size for regions is a property enforced at the other levels by creating a higher level merging c adjacent regions. This process creates a hierarchy of multilevel networks for a path finding search. The parameters $\langle r, m, c\rangle$ completely define the structure of a simple hierarchy of networks.

Given a hierarchy of networks, a hierarchical shortest path algorithm starts at the lowest level network from the source vertex. When the current region is completely traversed, the search is promoted to the next higher level. The promotion step is executed once at each level until it reaches the highest level. At this point, the search is demoted to the next lower level towards the destination vertex. The demotion step is performed until encountering the lowest level network. The resulting *hierarchical shortest path* consists of subpaths at each level of the network. If we seek only the shortest path cost, the hierarchical path cost is enough. However, when the actual path is required, the hierarchical approach executes shortest path queries inside individual regions, denoted here as *intra-regional queries*, for each level in order to expand these subpaths to the next lower level. These intra-regional queries are performed until the whole path is represented by subpaths at the lowest level network.

In a hierarchical approach, the shortest subpaths found at higher levels are expanded towards lower levels through intra-regional queries. Formally, an intra-regional shortest path query concerns the computation of a shortest path inside one region between two vertices on the border of the region. Therefore, these intra-regional queries are an essential component of a hierarchical approach for the computation of shortest paths in geographical networks. The subdivision of large geographical networks is not restricted to hierarchical approaches and, consequently, intra-regional queries are used in other frameworks. As an example, the TIGER files of the U.S. Census Bureau are subdivided into counties.

In this paper, we propose a new strategy to execute intra-regional shortest path queries at the lowest level of a hierarchical network. This strategy is based on a hybrid shortest path algorithm that uses both a partial path view and a multiple-source shortest path algorithm. A partial path is a pre-computed subsequence of vertices that belong to the shortest path. The partial path is used to find a single shortest path between two vertices on the border of a region. The novel contribution of our paper is this new strategy for intra-regional shortest path queries.

Using our hybrid shortest path algorithm, the time and space requirements for intra-regional queries at the lowest level of the hierarchical network is $O(m)$ assuming a uniform distribution of vertices. This is an improvement compared to the $O(m^{1.5})$ space required by a path view algorithm. The time efficiency is achieved by extending a shortest path algorithm to consider pre-computed guide vertices and to visit a much smaller number of vertices in the process.

The rest of this paper is organized as follows. In Sect. 2, we review work related to intra-regional queries. Hierarchical approaches are described in Sect. 3. We present our hybrid shortest path algorithm in Sect. 4. Section 5 presents our experimental results. Section 6 discusses the main contributions of this paper.

2 Related Work

This work is a natural progression from our prior work on building systems to support both feature-based queries (“Where is X happening?”) and location-based queries (“What is happening at location Y?”) (Aref and Samet 1990) as in systems such as QUILT (Shaffer et al. 1990) and the SAND Browser (Samet et al. 2003). It is also applicable to surface data (e.g., Sivan and Samet 1992). Queries on road networks have received particular interest (Sankaranarayanan and Samet 2010b) with shortest path finding receiving renewed attention due to applications in spatial network databases such as MapQuest, Google Maps, Yahoo! Maps, Bing Maps, and others. Among these applications, nearest neighbors algorithms (Chen et al. 2009; Demiryurek et al. 2009) require the efficient computation of shortest path distances in spatial networks. However, besides these client applications, shortest path finding has been recently addressed with regards to the efficient encoding of path views.

Samet et al. (2011), Sankaranarayanan et al. (2005, 2006) pre-compute the shortest paths between all possible vertices in the network. The path view is encoded by subdividing the shortest paths from a single vertex based on the first edges of each shortest path. They further reduce the space requirements to store the path view by exploring spatial coherence with a shortest path quadtree. Similarly, Sankaranarayanan and Samet (2009, 2010a); Sankaranarayanan et al. (2009) propose a new encoding of the path view that aggregates source and destination vertices into groups that share common vertices or edges on the shortest paths between them.

Frederickson (1987) proposed a hierarchical algorithm for the single-source shortest path problem on planar graphs. The shortest paths between every pair of border vertices are found for two levels. Therefore, the algorithm uses a path view in the search for a shortest path through the hierarchy of networks.

Jing et al. (1998) suggested a path view approach that stores the direct successor vertex and the cost of a shortest path for each source-destination pair in a region. Therefore, the path view requires $O(m^2)$ space for a region at the lowest level, where m is the number of vertices in a single region. They use a path finding algorithm that recursively queries the shortest path cost through all levels in the hierarchy of networks. They first determine the sets B_s and B_d of border vertices in regions containing the source vertex s and the destination vertex d, respectively. The algorithm uses pre-computed shortest paths between s and B_s; B_s and B_d; and B_d and d to find the global minimum cost for the path from s to d by searching among all pairs (b_s, b_d) of border vertices, where $b_s \in B_s$ and $b_d \in B_d$. The algorithm does not compute the whole path described by edges at the lowest level.

Shekhar et al. (1997) focus on path view implementations for a two-level hierarchy. They proposed a hybrid path view that encodes the direct successor and cost for any shortest path only from interior vertices to the border vertices in each lowest level region. The higher level is fully materialized. Grid graphs were used in a complexity analysis of the space required for the path views. The space storage required for the path views is $O(n^{5/3})$. In this paper, we present a hybrid path view that requires $O(n)$ space based on partial paths.

Goldberg and Harrelson (2005) propose a flat shortest path algorithm that prunes the number of visited vertices as does our hybrid shortest path algorithm for the lowest level of the hierarchy. They use A^* search with cost bounds computed according to the triangle inequality and distances between sampled (possibly random) vertices called landmarks.

3 A Hierarchical Approach

A *hierarchical approach* is based on the subdivision of the original network into regions. A region corresponds to a connected subgraph of the graph representing the network. A higher level network consists only of border vertices. A *border vertex* is a vertex that belongs to at least two different regions in the network. Since a shortest path passing through more than one region must include border vertices, the edges of a higher network represent possible connections between border vertices in this network.

The 0-level network is the original network represented by an embedding of an undirected planar graph $G(V, E)$ on the plane, where V (E) is the set of vertices (edges) in G. The number of vertices in V is n. This graph is subdivided into r smaller connected regions corresponding to subgraphs $\langle G_0^0, G_0^1, \ldots, G_0^{r-1} \rangle$ such that these subgraphs cover the original network ($V = V_0^0 \cup V_0^1 \cup \cdots \cup V_0^{r-1}$, $E = E_0^0 \cup E_0^1 \cup \cdots \cup E_0^{r-1}$) and each edge belongs to only one subgraph.

Each 0-level subgraph has m vertices and forms a suitable subdivision of the graph G where boundaries of each corresponding region has a size of $O(\sqrt{m})$ vertices. Such suitable subdivision is obtained in $O(n \log n)$ time using a fragmentation algorithm (Frederickson 1987). Goodrich (1992) proposed an algorithm to find a separator decomposition and a suitable subdivision in $O(n)$ time.

For $k > 0$, the k-level network is generated from the $(k-1)$-level network. A vertex v is in the k-level network if it belongs to two different $(k-1)$-level regions. Note that the k-level network has only border vertices. There is an edge connecting two k-level vertices in the k-level network if there is a path connecting them in the same $(k-1)$-level region. The k-level network is subdivided into connected k-level regions containing c adjacent $(k-1)$-level regions such that each $(k-1)$-level region belongs to only one k-level region.

In a hierarchy of networks, the graph G associated with the original network is represented by a set G_0 of r subgraphs $\langle G_0^0, G_0^1, \ldots, G_0^{r-1} \rangle$. These subgraphs correspond to regions that will be merged into higher level regions containing c regions.[1] This way, all regions at a particular level have about the same number of vertices. Each subgraph is denoted by $G_k^i \left(V_k^i, E_k^i\right)$, where k is the level in the hierarchy and i represents a region. The set of subgraphs representing regions for a k-level is denoted by G_k. The set of border vertices in a subgraph G_k^i is represented by B_k^i and B_k denotes the border vertex set for the k-level.

PV_k^i is the path view for the border vertices in G_k^i. The path view for the border vertices contains the information required to retrieve a shortest path only between any pair of border vertices in a k-level region i. The function $PV_k^i(u, v)$ returns the predecessor of vertex v in the shortest path from vertex u. L_k^i is the set of edges linking all pairs of border vertices that are connected by a path in the subgraph G_k^i and L_k denotes these sets for the k-level. Each edge $(u, v)_i$ in L_k^i represents a shortest path from u to v in G_k^i. If there is another edge $(u, v)_j$ in L_k, then we just retain the edge with the minimum cost. The relation T in a particular k-level network represents the topological relation between k-level regions such that $(i, j) \in T$ if and only if the intersection set of vertices $B_k^{i,j} = B_k^i \cap B_k^j$ is not empty, where i and j are indices for different k-level regions.

[1] We assume without loss of generality that r is a power of c, i.e., $r = c^h$, where h is the highest level in the hierarchy of networks.

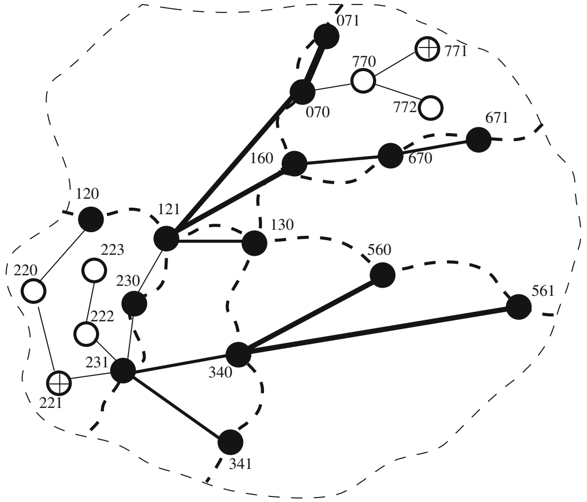


Fig. 1 Shortest path tree layers in a network subdivided into six regions. The source and destination vertices are depicted with a *cross* inside a *circle*. Border vertices are shown as filled *black circles*. Only the regions containing the source and destination vertices are depicted with all vertices. The other regions are shown only through border vertices

Lemma 1 *Since the number of vertices embedded in a k-level region corresponding to subgraph G_k^i is $O(c^k m)$, the number of border vertices $|B_k|$ is $O\left(\sqrt{c^k m}\right)$, where m is the number of vertices in a 0-level subgraph (*Frederickson 1987; Goodrich 1992; Lipton and Tarjan 1979*).*

Lemma 2 *If $k > 0$, the number of vertices $|V_k|$ in a k-level subgraph G_k^i is $O\left(\sqrt{c^{k+1} m}\right)$ and the number of edges $|E_k|$ is $O\left(c^k m\right)$.*

A hierarchical shortest path algorithm creates shortest path tree layers PT_k^+ (for promotion) and PT_k^- (for demotion) at each k-level of the hierarchy $\langle G_0, G_1, \ldots, G_h \rangle$ of networks, where G_k represents a level $\langle G_k^0, G_k^1, \ldots, G_k^{r_k-1} \rangle$ in the hierarchy. We define a *shortest path tree* as a tree whose unique simple path from root to any vertex represents a shortest path. A *layer* of a shortest path tree is a subset of a shortest path tree contained in a particular level of the hierarchy of networks (see Fig. 1). The shortest path tree layers are computed in order to find the *hierarchical shortest path* from a source vertex s to a destination vertex d throughout all levels of the hierarchy. The algorithm uses a set S that represents source vertices for a layer at a k-level, where each vertex in S is associated with a cost.

3.1 Expanding Subpaths in Higher Levels

A *hierarchical shortest path* from s to d consists of a sequence of subpaths $\langle P_0^+(s, v_1), P_1^+(v_1, v_2), P_2^+(v_2, v_3), \ldots, P_{h-1}^+(v_{h-1}, v_h), P_h^-(v_h, v_{h+1}),$ $P_{h-1}^-(v_{h+1}, v_{h+2}), \ldots, P_2^-(v_{2h-2}, v_{2h-1}), P_1^-(v_{2h-1}, v_{2h}), P_0^-(v_{2h}, d)\rangle$, where a subpath from vertex v_i to vertex v_j in the k-level network is denoted by either $P_k^+(v_i, v_j)$ or $P_k^-(v_i, v_j)$. In order to have a complete shortest path, a hierarchical algorithm executes intra-regional queries for each k-level expanding subpaths at the k-level to subpaths at the next lower level until the whole path consists only of edges in the lowest level network.

The `Expand-Path` algorithm given below finds a whole shortest path from s to d represented by a sequence of edges in the lowest level network. The algorithm traverses the shortest path tree layers $PT_k^{\pm}$ in order to retrieve the subpaths that compose a shortest path from s to d at all levels of the hierarchy. Then, the algorithm performs intra-regional queries to expand each edge of these subpaths into a path P at a lower level.

```
Algorithm Expand-Path($\langle PT_0^+, \ldots, PT_{h-1}^+, PT_h^-, PT_{h-1}^-, \ldots, PT_0^- \rangle, s, d$)
 1. $P_0^-(v_{2h}, d) \leftarrow$ Traverse-Backwards$(PT_0^-, d)$
 2. for $k \leftarrow 1$ to $h$ do
    (a) $P_k^-(v_{2h-k}, v_{2h-k+1}) \leftarrow$ Traverse-Backwards$(PT_k^-, v_{2h-k+1})$
 3. for $k \leftarrow h - 1$ to $1$ do
    (a) $P_k^+(v_k, v_{k+1}) \leftarrow$ Traverse-Backwards$(PT_k^+, v_{k+1})$
 4. $P_0^+(s, v_1) \leftarrow$ Traverse-Backwards$(PT_0^+, v_1)$
 5. for $k \leftarrow h$ to $1$ do
    (a) $P \leftarrow \emptyset$
    (b) for $u \leftarrow v_k \wedge (u, v)_i \in P_k^-(v_k, v_{2h-k+1})$ to $v_{2h-k+1}$ do
          i. $P \leftarrow P \oplus$ Intra-Regional$(u, v, k, i)$
    (c) $P_{k-1}^-(v_{k-1}, v_{2h-k+2}) \leftarrow P_{k-1}^+(v_{k-1}, v_k) \oplus P \oplus P_{k-1}^-(v_{2h-k+1}, v_{2h-k+2})$
```

Initially, the algorithm traverses each shortest path tree layer backwards using the procedure `Traverse-Backwards` (steps 1, 2, 3, and 4). This procedure creates the subpaths $P_k^+(v_j, v_{j+1})$ and $P_k^-(v_j, v_{j+1})$ at each k-level by retrieving a sequence of edges $(u, v)_i$ for each subpath. The algorithm expands each subpath starting at the highest level until it finds a whole path represented by edges only at the lowest level (step 5). For each k-level, the algorithm takes the corresponding subpath $P_k^-(v_k, v_{2h-k+1})$ and the procedure `Intra-Regional` expands each edge $(u, v)_i$ in this subpath into a subpath in the $(k - 1)$-level (step 5.b). This procedure finds the subpath from u to v in the subgraph G_{k-1}^i corresponding to a region i at the $(k - 1)$-level. These subpaths at the $(k - 1)$-level are concatenated (operator $\oplus$) into a path P (step 5.b.i). Then, all subpaths at the $(k - 1)$-level are concatenated into only one subpath $P_{k-1}^-(v_{k-1}, v_{2h-k+2})$ (step 5.c).

An important issue in the complexity analysis of the `Expand-Path` algorithm is the number of edges that an expanded shortest path will have at a particular level of the hierarchy.

Lemma 3 *The number of edges in a shortest path expanded to the $(h-i)$-level is $O(2^i)$.*

For intra-regional queries, we may use any flat strategy. However, we propose an efficient method for this task in the next section. A path view strategy pre-computes all shortest paths in the network for each region. When $c = 2$, this strategy requires $O(m)$ time to retrieve a shortest path at the lowest level and $O(1)$ time otherwise. The path view strategy requires $O(m^{1.5})$ space at the lowest level and $O(|B_k|^2) = O(c^k m)$ otherwise. The path view at the highest level and the expanded shortest path at the lowest level require $O(n)$ space. They dominate the space requirement for `Expand-Path`.

Theorem 1 *If the intra-regional query is implemented as a path view approach, the algorithm* `Expand-Path` *requires $O(n)$ time.*

The path view in the highest level and the expanded shortest path in the lowest level require $O(n)$ space. They dominate the space requirement for the algorithm `Expand-Path`.

Theorem 2 *If the intra-regional query is implemented as a path view approach, the algorithm* `Expand-Path` *requires $O(n + m^{1.5})$ space.*

Another flat strategy for intra-regional queries is a single-source shortest path algorithm (Dijkstra 1959). This strategy needs $O(|V_k| \log |V_k| + |E_k|)$ time to find a shortest path in a k-level subgraph and requires $O(|V_k| + |E_k|)$ space.

Theorem 3 *If the intra-regional query is implemented as a single-source shortest path algorithm, the algorithm* `Expand-Path` *requires $O(n \log m)$ time.*

The single-source shortest path algorithm requires space for two subgraphs at each level of the hierarchy (except the highest) and for the expanded path P. The total space required is $O(n)$. Therefore, there is no improvement concerning space requirement when a single-source shortest path algorithm performs intra-regional queries in a hierarchical approach.

Theorem 4 *If the intra-regional query is implemented as a single-source shortest path algorithm, the algorithm* `Expand-Path` *requires $O(n)$ space.*

4 The Hybrid Shortest Path Algorithm

Given an edge $e = (u, v)_i$ at the k-level network ($k > 0$), an intra-regional query consists of expanding the edge e into a subpath from u to v in the subgraph G^i_{k-1} at the $(k-1)$-level, where u and v are border vertices in G^i_{k-1}. The query may be performed using any flat strategy, ranging from a single-source shortest path algorithm to a lookup in a path view for border vertices.

The strategy proposed in this paper uses a hybrid path view for border vertices. The hybrid path view represents each shortest path between border vertices of a

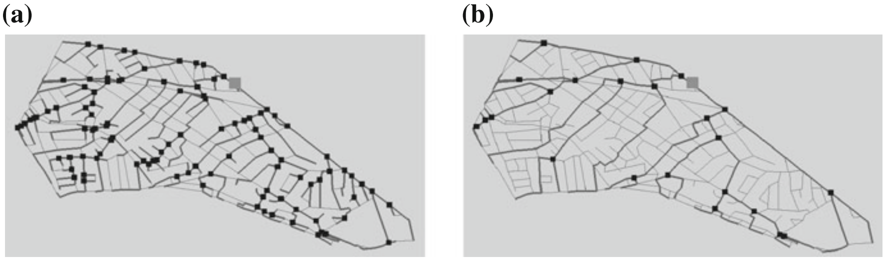


Fig. 2 Shortest path tree from a source vertex (*gray square*) and guide vertices (*black squares*) in a lowest level region. **a** Destinations are all vertices. **b** Destinations are only border vertices

region by just a sequence of guide vertices. A usual path view is implemented as a predecessor matrix, where each row of the predecessor matrix represents a shortest path tree corresponding to a particular source vertex as the root. Each entry of the matrix row specifies the predecessor vertex for a vertex in the shortest path tree. A *guide vertex* is a vertex acting as predecessor vertex for more than one vertex in the predecessor matrix implementing the path view for border vertices. The path view for border vertices consists of shortest path trees composed only of shortest paths from a border vertex to the other border vertices (see Fig. 2).

Lemma 4 *The number of guide vertices in a shortest path tree of a path view for a subgraph at the k-level is* $O(\sqrt{c^k m})$.

The hybrid path view $\overline{PV^i_k}$ for each region i at each k-level of the network hierarchy is retrieved from the corresponding path view PV^i_k for border vertices. In order to create $\overline{PV^i_k}$, each guide vertex is identified in a traversal of PV^i_k. A hybrid path view is implemented as a predecessor sparse matrix whose columns only consider guide vertices $\overline{V^i_k}$ and border vertices B^i_k. Furthermore, the predecessor relation in this sparse matrix is only expressed in terms of guide vertices.

The algorithm `Hybrid-View` below creates a hybrid path view $\overline{PV^i_k}$ from the path view PV^i_k. Each path view PV^i_k is associated with a set of vertices V^i_k and a set of border vertices B^i_k. The algorithm builds a matrix Γ to keep track of the shortest path tree considering only paths to border vertices. The algorithm uses a vector Λ to store the number of times any vertex $v \in V^i_k$ is a predecessor in a particular shortest path tree of PV^i_k.

Algorithm `Hybrid-View`(PV^i_k)

```
1. for $u \in B^i_k$ do
   (a) $\overline{V^i_k} \leftarrow B^i_k$
   (b) $\Lambda \leftarrow 0$
   (c) $\Gamma \leftarrow$ false
   (d) for $v \in B^i_k$ do
       i. $\Gamma(u, v) \leftarrow$ true
```

```
        ii. $v' \leftarrow PV_k^i(u, v)$
        iii. **while** $v' \neq u$ **do**
            A. $\Gamma(u, v') \leftarrow$ **true**
            B. $v' \leftarrow PV_k^i(u, v')$
    (e) **for** $v \in V_k^i$ **do**
        i. **if** $\Gamma(u, v)$ **then**
            A. $\Lambda(PV_k^i(u, v)) \leftarrow \Lambda(PV_k^i(u, v)) + 1$
    (f) **for** $v \in V_k^i$ **do**
        i. **if** $\Lambda(v) > 1$ **then**
            A. $\overline{V_k^i} \leftarrow \overline{V_k^i} \cup v$
    (g) **for** $v \in \overline{V_k^i}$ **do**
        i. $v' \leftarrow PV_k^i(u, v)$
        ii. **while** $v' \notin \overline{V_k^i}$ **do**
            A. $v' \leftarrow PV_k^i(u, v')$
        iii. $\overline{PV_k^i}(u, v) \leftarrow v'$
```

Hybrid-View algorithm traverses the path view PV_k^i computing the hybrid path for each vertex $u \in B_k^i$ corresponding to a shortest path tree in PV_k^i. Initially, the algorithm finds the shortest path tree considering only paths to border vertices (step 1.d). The matrix Γ identifies the edges belonging to this shortest path tree (see Fig. 2b). According to Γ, the algorithm computes the vector Λ that stores the number of sons for each vertex in the shortest path tree considering only paths to border vertices (step 1.e). Each vertex $v \in V_k^i$ that is a predecessor for more than one vertex in a shortest path tree is considered a guide vertex and inserted in $\overline{V_k^i}$ (step 1.f). Next, the algorithm computes the corresponding shortest path tree only in terms of guide vertices in $\overline{V_k^i}$ (step 1.g). The predecessor of a vertex in $\overline{V_k^i}$ is the first vertex in a backward traversal of the corresponding shortest path tree in PV_k^i that belongs to $\overline{V_k^i}$.

The hybrid path view computation implies pre-processing time. The time required for the Hybrid-View algorithm is dominated by the path view scanning that counts predecessors and finds guide vertices.

Theorem 5 *Algorithm* Hybrid-View *needs $O(m^2)$ time at the lowest level and $O((c^k m)^{1.5})$ time otherwise.*

The hybrid path view computation at all levels of the hierarchy requires additional pre-processing time $O(rm^2 + \sum_{k=1}^{h} \frac{r}{c^k}(c^k m)^{1.5})$. This expression is equivalent to $O(nm + n\sqrt{n}) = O(n(m + \sqrt{n}))$. If we assume $r = m = \sqrt{n}$, then the additional pre-processing time becomes $O(n^{1.5})$.

At higher levels, a hybrid path view has the same space requirements as a path view when $c = 2$. However, at the lowest level, a hybrid path view requires $O(m)$ space. This is an improvement compared to the $O(m^{1.5})$ space required by a path view at this level. This way, the Hybrid-View algorithm space requirements are dominated by the path view predecessor matrix.

Theorem 6 *Algorithm* `Hybrid-View` *requires* $O(m^{1.5})$ *space at the lowest level and* $O(c^k m)$ *space otherwise.*

An intra-regional query for an edge $(u, v)_i$ in a 1-level network is performed by the hybrid shortest path algorithm. This algorithm uses the subgraph G_0^i and the hybrid path view $\overline{PV_0^i}$ information in order to find a single shortest path P from u to v in the 0-level region i. $\overline{PV_0^i}$ describes the shortest path in terms of guide vertices as a partial shortest path $\overline{P_0^i}(u, v)$. Therefore, the algorithm expands $\overline{P_0^i}(u, v)$ finding a sequence of subpaths between consecutive guide vertices in $\overline{P_0^i}(u, v)$.

The algorithm `Hybrid-Shortest-Path` below creates a shortest path tree PT whose root represents the source vertex u. A priority queue Q is used to keep track of all information related to the current vertices in $V_{k-1}^i - PT$. Each entry $(u, f(u), e)$ in Q represents a vertex u with an estimate cost $f(u)$ and a predecessor edge e. The set Q' keeps track of the vertices in Q whose cost is not infinity and, consequently, will be reset for the next subpath search.

Algorithm `Hybrid-Shortest-Path`(u, v, k, i)

```
1. $\overline{P_0^i}(u, v) \leftarrow$ Path-Lookup$\left(\overline{PV_0^i}, u, v\right)$
2. $PT \leftarrow Q \leftarrow \emptyset$
3. $Q \leftarrow Q \cup (u, 0, 0)$
4. for $v' \in V_0^i \wedge v' \neq u$ do
   (a) $Q \leftarrow Q \cup (v', \infty, \infty)$
5. for $(s, d) \leftarrow (u, v') \wedge (s, d) \in \overline{P_0^i}(u, v)$ to $(v'', v)$ do
   (a) $(u', f(u'), e') \leftarrow$Extract-Min($Q$)
   (b) $Q' \leftarrow Q' - u'$
   (c) $PT \leftarrow PT \cup (u', f(u'), e')$
   (d) while $u' \neq d$ do
       i. for $(u', u'')_j \in E_0^i$
          A. if $f(u') + |(u', u'')_j| < f(u'')$ then
             – $f(u'') \leftarrow f(u') + |(u', u'')_j|$
             – $e'' \leftarrow (u', u'')_j$
             – $Q' \leftarrow Q' \cup u''$
       ii. $(u', f(u'), e') \leftarrow$Extract-Min($Q$)
       iii. $Q' \leftarrow Q' - u'$
       iv. $PT \leftarrow PT \cup (u', f(u'), e')$
   (e) for $u' \in Q'$ do
       i $f(u') \leftarrow e' \leftarrow \infty$
   (f) for $(d, u'')_j \in E_0^i$
       i. $f(u'') \leftarrow |(d, u'')_j|$
       ii. $e'' \leftarrow (d, u'')_j$
6. $P \leftarrow$Traverse-Backwards($PT, v$)
```

First, the algorithm finds the partial shortest path $\overline{P_0^i}(u, v)$ using the procedure `Path-Lookup` which traverses the hybrid path view $\overline{PV_0^i}$ (step 1). Initially, PT is empty and Q has all vertices with cost and predecessor edge equal to ∞, but the source vertex u whose cost is 0 (steps 2, 3, and 4). Next, the algorithm finds a shortest

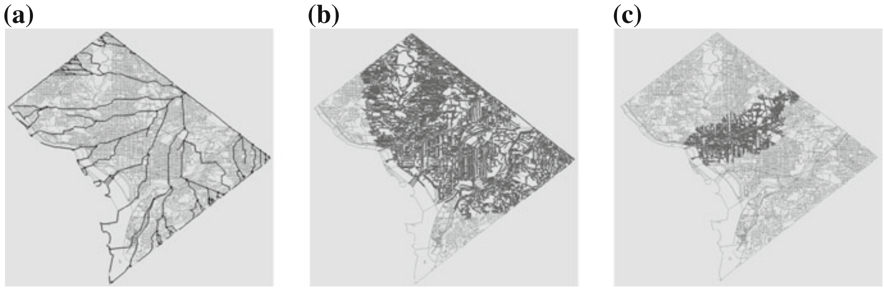


Fig. 3 Vertices visited by shortest path algorithms. **a** Shortest path tree for border vertices. **b** Dijkstra's algorithm. **c** Our hybrid algorithm

subpath in G_0^i corresponding to each edge (s, d) in the partial shortest path (step 5). The shortest subpath for each edge is computed in a similar way to Dijkstra's shortest path algorithm. However, the current state of the priority queue Q means that there is no need to consider the vertices already in PT for the current subpath. Therefore, each subsequent search has an initial Q just with all remaining vertices (step 5.e). All costs and predecessor edges are set to ∞, but vertices connected to d have its costs and predecessor edges updated to represent that the next source vertex is d with cost equal to 0 (step 5.f). The subpath from u to v is computed by traversing PT from v using the procedure `Traverse-Backwards` (step 6).

In the worst case, the hybrid shortest path algorithm has the same time complexity as Dijkstra's single-source shortest path algorithm (Dijkstra 1959). However, the worst case only happens when all vertices of the subgraph are visited by the algorithm. The number of vertices visited by the hybrid shortest path algorithm is much smaller than the number of vertices of the subgraph when vertices are uniformly distributed in the region corresponding to the subgraph and the guide vertices are uniformly distributed along a shortest path (see Fig. 3).

Theorem 7 *Assume a uniform distribution of vertices at the lowest level and the guide vertices are also uniformly distributed along a shortest path. The number of vertices visited by the hybrid shortest path algorithm is* $O(\sqrt{m})$.

Assuming that vertices are uniformly distributed, the running time for the hybrid shortest path algorithm becomes $O(m)$ at the lowest level.

Theorem 8 *Assume a uniform distribution of vertices at the lowest level and the guide vertices are also uniformly distributed along a shortest path. The hybrid shortest path algorithm spends* $O(m)$ *time at the lowest level.*

Since the subgraph G_0^i and the hybrid path view $\overline{PV_0^i}$ are the inputs for the hybrid shortest path algorithm, the space requirements for the algorithm are the same as for a single-source shortest path algorithm.

Theorem 9 *The hybrid shortest path algorithm requires $O(m)$ space in the k-level when $k = 1$ and $O(c^{k-1}m)$ space otherwise.*

We have now seen that the best time and space requirements for intra-regional queries at the lowest level are achieved by the hybrid shortest path algorithm. On the other hand, for higher levels, the best strategy concerning time and space requirements is the path view approach when $c = 2$. Therefore, the expansion of the hierarchical path into a path with edges at the lowest level requires $O(n)$ time and space.

5 Experimental Results

We evaluate the time and space performance of our hybrid approach using road networks from the TIGER files of the U.S. Census Bureau. We compare our method with two state-of-the-art techniques for shortest path finding on spatial networks: the shortest path quadtrees (Samet et al. 2011) and a flat approach based on Dijkstra's algorithm (Dijkstra 1959). All algorithms were implemented using C++ in a Dell XPS 730X machine with a i7 CPU at 3.20 GHz and 6 Gb of RAM. All data structures necessary for the execution of all algorithms were loaded in the main memory.

The implementation of our approach consists of four pre-processing modules: network, border, view, and hierarchy. They pre-process the TIGER/Line files in order to find the information required by the shortest path algorithms. A road network for each county in a state is obtained in the network module. After that, the border module finds for each county the set of border vertices that are shared with another county. Then, the view module computes the path view for each county. Finally, the hierarchy module finds a higher-level network for the state to be used in a hierarchical shortest path algorithm.

In our experimental setup, we initially selected a spatial network that requires only the amount of available space for all evaluated techniques. To obtain networks of smaller sizes, we determined the centroid of the original network and incrementally removed vertices according to their distance to the centroid. This way, we were able to generate networks of continuous sizes to evaluate the space requirements of all considered methods for a continuous range of network sizes. Similarly, for each network size, we compute shortest paths from the centroid vertex to the most distant vertex in the spatial network. This way, we investigated the time needed to found shortest paths of varying lengths. The memory space and the running time obtained in our experiments is shown in Fig. 4. The space requirements for our hybrid approach a virtually the same as the requirements of the Dijkstra's algorithm. The shortest path quadtree spends more space as expected from its space complexity $O(n^{1.5})$. In terms of running time, the hybrid approach performs best followed closely by the shortest path quadtree method.

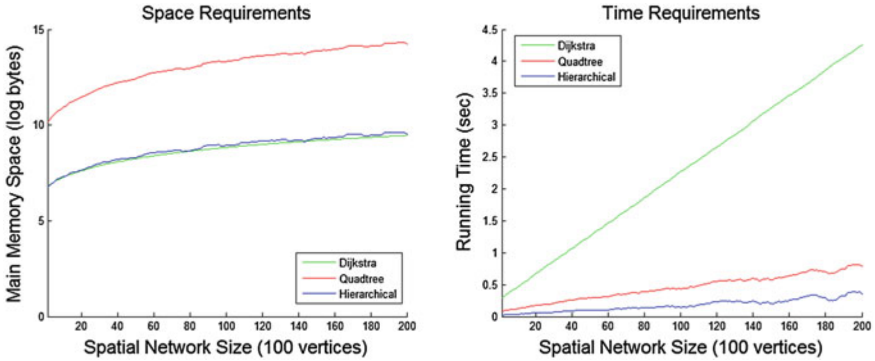


Fig. 4 The memory space and running time of our hybrid approach compared to other path finding methods

6 Conclusion

We introduced a hybrid shortest path algorithm to perform intra-regional queries. This strategy uses a subsequence of pre-processed vertices that belong to the shortest path while actually computing the whole shortest path. At the lowest level, the hybrid algorithm requires $O(m)$ time and space assuming a uniform distribution of vertices. For higher levels, the path view approach takes $O(1)$ time and requires $O(c^k m)$ space.

References

Aref WG, Samet H (1990) Efficient processing of window queries in the pyramid data structure. In: Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems (PODS), Nashville, pp 265–272

Chen Z, Shen H, Zhou X, Yu J (2009) Monitoring path nearest neighbor in road networks. In: Proceedings of the 35th SIGMOD international conference on management of data (SIGMOD'09), pp 591–602

Demiryurek U, Banaei-Kashani F, Shahabi C (2009) Efficient continuous nearest neighbor query in spatial networks using Euclidean restriction. In: Proceedings of the 11th international symposium on advances in spatial and temporal databases (SSTD'09), Lecture notes in computer science, vol 5644. Springer, Heidelberg pp 25–43

Dijkstra E (1959) A note on two problems in connection with graphs. Numer Math 1:269–271

Frederickson G (1987) Fast algorithms for shortest paths in planar graphs, with applications. SIAM J Comput 16(6):1004–1022

Goldberg A, Harrelson C (2005) Computing the shortest path: a^* search meets graph theory. In: Proceedings of the 16th annual ACM-SIAM symposium on discrete algorithms, pp 156–165

Goodrich M (1992) Planar separators and parallel polygon triangulation. In: Proceedings of the 24th ACM symposium on theory of computing, pp 507–516

Henzinger M, Klein P, Rao S, Subramanian S (1997) Faster shortest-path algorithms for planar graphs. J Comput Syst Sci 55(1):3–23

Jing N, Huang YW, Rundensteiner E (1998) Hierarchical encoded path views for path query processing: an optimal model and its performance evaluation. IEEE Trans Knowl Data Eng 10(3):409–432

Kolahdouzan, M, Shahabi C (2004) Voronoi-based *k* nearest neighbor search for spatial network databases. In: Proceedings of the 30th international conference on very large databases, Toronto, pp 840–851

Lipton R, Tarjan R (1979) A separator theorem for planar graphs. SIAM J Appl Math 36:177–189

Samet H, Alborzi H, Brabec F, Esperança C, Hjaltason GR, Morgan F, Tanin E (2003) Use of the SAND spatial browser for digital government applications. Commun ACM 46(1):63–66

Samet H, Sankaranarayanan J, Alborzi H (2011) Scalable network distance browsing in spatial databases. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data (SIGMOD'08), pp 43–54

Sankaranarayanan J, Samet H (2009) Distance oracles for spatial networks. Shanghai, pp 652–663

Sankaranarayanan J, Samet H (2010) Query processing using distance oracles for spatial networks. 22(8):1158–1175, best Papers of ICDE 2009 Special Issue

Sankaranarayanan J, Samet H (2010) Roads belong in databases. 33(2):4–11, invited paper

Sankaranarayanan J, Alborzi H, Samet H (2005) Efficient query processing on spatial networks. In: Proceedings of the 13th ACM international symposium on advances in geographic information systems, Bremen, pp 200–209

Sankaranarayanan J, Alborzi H, Samet H (2006) Distance join queries on spatial networks. In: Proceedings of the 14th ACM international symposium on advances in geographic information systems, Arlington, pp 211–218

Sankaranarayanan J, Samet H, Alborzi H (2009) Path oracles for spatial networks. In: Proceedings of the VLDB endowment, Maynooth, pp 1210–1221

Shaffer CA, Samet H, Nelson RC (1990) QUILT: a geographic information system based on quadtrees 4(2), 103–131, also University of Maryland Computer Science Technical Report TR-1885

Shekhar S, Fetterer A, Goyal B (1997) Materialization trade-offs in hierarchical shortest path algorithms. In: Scholl M, Voisard A (eds) Proceedings of the 5th international symposium on advances in spatial databases (SSD'97), Lecture notes in computer science, vol 1262. Springer, pp 94–111

Sivan R, Samet H (1992) Algorithms for constructing quadtree surface maps. In: Proceedings of the 5th international symposium on spatial data handling. vol 1. Charleston, pp 361–370

A Formalization of Topological Relations Between Simple Spatial Objects

Gutemberg Guerra-Filho, Claudia Bauzer Medeiros and Pedro J. de Rezende

Abstract This paper presents a new framework for modeling topological relations among objects of type point, line, and region. The main contributions are in two directions: First, the formalism proposed allows specifying all possible relations by means of symmetric matrices (whereas the usual formulation of such relations does not have this property). Symmetric matrices enable the efficient and automatic verification of valid matrices associated only with the possible topological relations. Second, it allows the specification of cases where two objects are spatially related in more than one way (*e.g.*, a line that crosses a given region in one part and is adjacent to the same region region in another part). This increases the flexibility that users are offered to model queries on spatial databases using topological relations.

Keywords Topological relations · 3-axis-Intersection · Conceptual neighborhood diagram

1 Introduction

Spatial relations play a central role in their (GIS) description into database query and spatial constructs, down to the query processing level (Clementini et al. 1992; Clementini et al. 1994). Most spatial query languages provide facilities and functions for expressing different spatial predicates, usually referring to topological and metric

Claudia Bauzer Medeiros is partially supported by grants from CNPq, CAPES and FAPESP. Pedro J. de Rezende is partially supported by CNPq grants 473867/2010-9, FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) grant 07/52015-0.

G. Guerra-Filho (✉)
Department of Computer Science and Engineering, University of Texas at Arlington,
Arlington, TX, USA
e-mail: guerra@cse.uta

C. B. Medeiros · P. J. de Rezende
Institute of Computing, UNICAMP, Brazil

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_5,

relations (Egenhofer 1991; Frank 1982; Herring et al. 1988; Raper and Bundock 1991; Roussopoulos et al. 1988). In order to support these operations, different formal frameworks have been introduced. Topological relations are the ones that have so far been studied the most. Direction relations have also been studied, and a recent formal model thereof can be found in Papadias et al. (1996).

In this paper, we present a new formal model for expressing topological relations among geometric objects. In particular, we focus on intersection matrices representations, which describe spatial relations in terms of the intersections of *interiors*, *boundaries*, and *exteriors* of objects. Existing matrix representations only allow for either the mere detection of empty intersection (*i.e.*, disjoint objects) or the enumeration of the dimension of intersections.

The various possible types of intersections between interiors, boundaries, and exteriors of geometric objects, once identified, are much more elucidative than simply knowing that an intersection exists or the set of intersection components' dimensions. In this sense, better models are needed to obtain a greater level of detail that allows for better distinction of seemingly identical topological relations making it possible for the user to focus on specific situations. However, this generality makes it necessary to define groups of relations that can be used to guide the user of the query system among the exceedingly large number of possibilities. Once such groups are well defined, one can take full advantage of the exact definitions of relations and consider all possible combinations of intersections. In this sense, Alboody et al. (2009) proposed detailed descriptions for four topological relations between regions in the intersection and difference model (Deng et al. 2007) using the separation number.

In this paper, we present a new formalism for topological relations where the (flat) intersection matrix representation is expanded into a (3-*d*) cube. The third and new dimension of this (3-*d*) cube, named the *3-axis-intersection model*, is the means for expressing and counting the ways in which two geometric objects are spatially related. A major contribution of this new framework is allowing the formal description of complicated scenarios. In these complicated scenarios, users are required to express when two objects maintain multiple different spatial relations at the same time (*e.g.*, a line that crosses a given region in one part and is adjacent to the same region in another part). Existing formalisms only describe a single topological relation at a time. In addition to that, our model allows to discriminate when two objects maintain the same topological relation multiple times by counting the number of such instances. We show that, by using our framework, one can determine beforehand an upper bound for all possible types of topological relations composed of combinations of multiple relations.

Our formalism uses definitions of topological parts (*i.e.*, interior, boundary, and exterior) for simple objects based on the topology of metric spaces. These definitions are slightly different from the ones used by previous models. Our approach leads to automatic methods to quantitatively compare different relations (*i.e.*, a topological distance). Based on this topological distance and on the ability to validate matrices, our approach allows the automatic computation of a conceptual neighborhood diagram.

A *conceptual neighborhood diagram* (Egenhofer and Mark 1995; Freksa 1992) is a graph where every node is a different topological relation and two nodes are linked if their topological distance is minimum. The quantitative comparison between different topological relations allows the automatic construction of conceptual neighborhood diagrams. Therefore, using our formalism, one can automatically establish a detailed conceptual neighborhood diagram. This lowest-level diagram can be partitioned into groups of similar topological relations. This process results into a generalized hierarchy of conceptual neighborhood diagrams, thereby helping users express their view of reality in more abstract ways and at different levels or granularities.

All existing formalizations are based on modelling topological relations between simple spatial objects. The applications of our approach also extends to consider relations among *complex* spatial objects—*e.g.*, a region composed of disjoint regions. The relationships between two complex objects are described in terms of the relations among their individual components.

Ultimately, our framework contributes to provide formal methods to naive geography, where users' intuitive descriptions can be mathematically modelled in a formal and compact notation. This notation enables the specification of queries according to topological relations and allows the search and retrieval of spatial and geographical entities in a database.

In summary, the main contributions of our new formalism for topological relations are (1) allowing the formal description of scenarios when two spatial objects have multiple relations at the same time, (2) enabling the automatic construction of conceptual neighborhood diagrams at different levels, (3) generalizing to complex spatial objects by considering the relations of individual components, and (4) providing formal methods to naive geography and to the specification of spatial queries.

The rest of this paper is organized as follows. Section 2 gives a brief overview of the main formal models that have been proposed to describe binary topological relations. Section 3 shows how these notions can be re-stated by using mathematical topology definitions. Section 4 extends this formalism by considering that two objects can simultaneously relate in several ways. Section 5 introduces the topological distance between relations and the construction of conceptual neighborhood diagrams. Finally, Sect. 6 presents conclusions and future work.

2 Related Work

In this section, we give an overview of the theoretical basis for defining binary topological relations and summarize matrix-based representations reported in the literature.

The *4-intersection* model is a widely accepted means for the representation of topological relations between region objects (Egenhofer and Herring 1990), in which the definition of relations between objects A and B is based on the four intersection sets of their interiors ($^\circ$) and boundaries (∂). The intersection sets are denoted by $S_{i,j}$, where i, j indicate the operands of the intersections as follows:

$$
\begin{aligned}
S_{0,0} &= A^{\circ} \cap B^{\circ}, \\
S_{0,1} &= A^{\circ} \cap \partial B, \\
S_{1,0} &= \partial A \cap B^{\circ}, \\
S_{1,1} &= \partial A \cap \partial B.
\end{aligned}
$$

A 2 × 2-matrix $\Im_4$ concisely represents these criteria (Eq. 1). These four intersection sets form a topological invariant of the relation between A and B. The set of values that represents the content of the intersection sets is denoted by *domain* $Dom(S)$, where S is an intersection set. The 4-intersection method regards only the values empty and non-empty as domain ($Dom_4(S) = \{\emptyset, \neg\emptyset\}$).

$$
\Im_4(A, B) = \begin{pmatrix} S_{0,0} & S_{0,1} \\ S_{1,0} & S_{1,1} \end{pmatrix} \tag{1}
$$

Egenhofer and Herring (1991a,b) introduced the *9-intersection* model extending the 4-intersection model to account for intersections between pairs of objects other than (2-d) regions, such as pairs of lines, or a line and a region. The 9-intersection model describes binary topological relations based on the intersections of the interiors, boundaries, and exteriors ($^-$) of two given spatial objects A and B:

$$
\begin{aligned}
S_{0,2} &= A^{\circ} \cap B^{-}, \\
S_{1,2} &= \partial A \cap B^{-}, \\
S_{2,0} &= A^{-} \cap B^{\circ}, \\
S_{2,1} &= A^{-} \cap \partial B, \\
S_{2,2} &= A^{-} \cap B^{-}.
\end{aligned}
$$

The nine intersections provide a formal description of the topological relations between the objects, which can be concisely represented by a 3 × 3-matrix $\Im_9$ (Eq. 2). This model applies the same domain of 4-intersection: $Dom_9(S) = Dom_4(S)$.

$$
\Im_9(A, B) = \begin{pmatrix} S_{0,0} & S_{0,1} & S_{0,2} \\ S_{1,0} & S_{1,1} & S_{1,2} \\ S_{2,0} & S_{2,1} & S_{2,2} \end{pmatrix} \tag{2}
$$

In the *dimension extended method* (Clementini and Felice 1995; Clementini et al. 1992), Clementini et al. take into account the highest dimension of the intersection, instead of only distinguishing the content (emptiness or non-emptiness) of the intersection. This method is also an extension of the 4-intersection model, where the intersection set S can now be either $\emptyset$, 0-d, 1-d, or 2-d: $Dom_{dim}(S) = \{\emptyset, 0{-}d, 1{-}d, 2{-}d\}$. For instance, the 4 intersections between a line and a region result in the following possible cases:

Table 1 The number of relations for all relation groups

Relation groups	4-int	9-int	ext-dim	ref-dim
Region/region	8	8	9	16
Region/line		19	17	43
Line/line		33	18	61
Region/point			3	3
Line/point			3	3
Point/point			2	2

$$\begin{aligned}
Dom_{dim}(S_{0,0}) &= \{\emptyset, 1\text{-}d\},\\
Dom_{dim}(S_{0,1}) &= \{\emptyset, 0\text{-}d\},\\
Dom_{dim}(S_{1,0}) &= \{\emptyset, 0\text{-}d, 1\text{-}d\},\\
Dom_{dim}(S_{1,1}) &= \{\emptyset, 0\text{-}d\}.
\end{aligned}$$

Based on the dimension extended model, Clementini et al. also show that, from the users' point of view, all binary relations can be expressed in terms of 5 operators (cross, in, overlap, disjoint, touch) and two boundary functions. This means that, in order to formulate queries or to describe a scenario, users can precisely express what they want using only this vocabulary.

McKenney et al. (2005) proposed the 9*-intersection dimension matrix* based on the dimension of an intersection as the topological invariant. In their model, the refined dimension of a given point set is the union of dimensions of its maximal connected components. Hence, the dimension matrix actually considers all possible dimension combinations of points, lines, and regions: {⊥, 0-*d*, 1-*d*, 2-*d*, 01-*d*, 02-*d*, 12-*d*, 012-*d*}, where ⊥ is the undefined dimension of an empty set, 0 is the dimension of single points, 1 is the dimension of single lines, and so on. Besides considering all possible combinations of dimensions in a more explicit manner where each unique dimension has its own entry in the matrix, our approach counts the number of components per intersection and for each dimension. This enables a higher level of discriminative power to represent more details in terms of topological relations. Furthermore, the basic predicates defining topological parts (*i.e.*, interior, boundary, exterior) are fundamentally different in our model. We follow a more strict topological definition.

Each matrix-based model allows representing a number of feasible binary topological relations among objects. These numbers are shown in Table 1. The table shows, for instance, that the 9-intersection model allows representing up to 8 different cases of relations among regions, and up to 33 cases of relations among lines.

The 9^{+}*-intersection model* (Kurata and Egenhofer 2007) considers the intersections of topological primitives of spatial objects. A topological primitive is a self-connected and mutually-disjoint subset of a topological part of the spatial object. Therefore, the 9^{+}-intersection model describes topological relations between complex objects that may consist of multiple disjoint subparts.

The *intersection and difference model* (Deng et al. 2007) uses only the interior and the boundary of regions. The model describes topological relations according to intersection sets ($A^\circ \cap B^\circ$ and $\partial A \cap \partial B$) and difference sets ($A - B$ and $B - A$) of two given spatial objects A and B.

Kurata (2009) proposed a method to build a conceptual neighborhood graph for a given set of topological relations according to the 9^+-intersection model. The graph links pairs of topological relations according to a smooth transformation that changes one relation into the other. Two topological relations r_a and r_b are *conceptual neighbors* when there exists a smooth transformation that changes the topological relation of two simple objects from r_a to r_b. Instead of using a smooth transformation, we construct conceptual neighborhood diagrams similarly to the snapshot model (Egenhofer and Al-Taha 1992). This approach considers similarity as the smallest number of different elements in a matrix-based representation. This allows the automatic computation of conceptual neighbors for a set of topological relations.

3 Simple Spatial Objects

The interior, boundary, and exterior are the *topological parts* of objects used in the literature to describe topological relations. Some of the previous representations are based on formal definitions for topological parts different from the pure mathematical theory (Alexandroff 1961). The difference lies in the definition of "boundary". First, unlike mathematical topology, points have no boundaries. Second, the boundary of a line is defined as being composed of two nodes at which exactly one 1-cell ends; its interior is the union of all interior nodes and all connections between the nodes.

Our definitions strictly follow the theory of general topology with a single definition of topological parts for all spatial objects. As some other previous models (Egenhofer and Franzosa 1991), our approach uses definitions of interior, boundary, and exterior for a simple object based on the standard general topology. These definitions are slightly different from the ones used by previous models. Here, we stress the differences between our work and the related literature with regards to the definitions concerning topological parts of simple spatial objects.

3.1 Definitions

To review the main definitions, we briefly recap a couple of them from topology of metric spaces. Let S be a subset of $\mathbb{R}^2$.

- A point $p \in S$ is an *interior point* of S if there is an open disc[1] centered at p totally contained in S. The set of all interior points of S is the *interior* of S—denoted S°.

[1] An open disc centered at a point p is the set $\{q \in \mathbb{R}^2 | d(p, q) < r\}$ for some positive radius r, where d is the Euclidean distance.

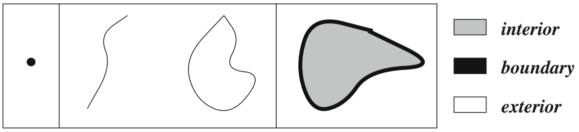


Fig. 1 Topological parts of simple objects

- A point $p \in \mathbb{R}^2$ is a *boundary point* of S if all discs of positive radius centered at p intersect both S and its complement (or *exterior*) $S^- = \mathbb{R}^2 - S$. The set of all boundary points of S is the *boundary* of S—denoted ∂S.
- S is said to be (topologically) *closed* if it contains its boundary.

From now on, the notions of interior, boundary, and exterior are to be regarded as those from traditional topology (Alexandroff 1961) whereby they are to be viewed in relation to the whole embedding topological space ($\mathbb{R}^2$), and not to a subspace of it.

We consider here any spatial objects in $\mathbb{R}^2$ of three possible dimensions, namely, (0-*d*) points, (1-*d*) curves, and (2-*d*) regions, provided that the last two satisfy certain conditions. A curve (which we often refer to as a *line*) must be a topologically closed arc of a simple Jordan curve of finite length. This means that it may have either two endpoints (included in the curve, in which case it is homeomorphic to a closed interval), or no endpoints (in which case it is homeomorphic to a circle—and is alluded to as a *cycle*). A region must be bounded and homeomorphic to a closed disc. Therefore, a line is connected, without self intersection, of finite length; and a region is topologically closed, connected, bounded, and simply-connected (without holes).

The interior of a point and that of a line are empty, while each of these is equal to its own boundary[2] (see Fig. 1). The interior of a region is homeomorphic to an open disc and its boundary is a line.

We will use the term *feature* to represent any spatial object of the types above. A *simple spatial object* obeys two properties:

a) it is (topologically) closed; and
b) it is connected, that is, it is not the union of two separated features.

4 A New Formalization for Spatial Relations

As shown in Sect. 2, previous formalizations of spatial relations between two objects are based on the specification of one relation at a time. This precludes the possibility of describing cases where two objects are related to each other in multiple ways. Figure 2 shows one such example: the line L both crosses and touches the region R.

[2] Some authors consider the boundary of a (non cycle) line as consisting of its two endpoints and its interior as the (non-empty) remaining arc.

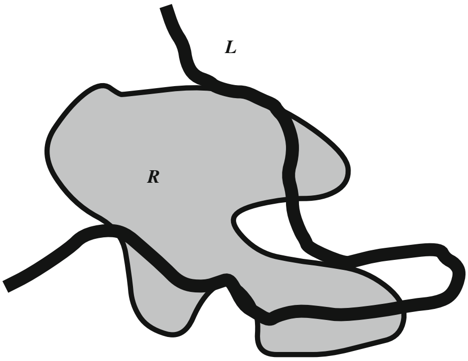


Fig. 2 A situation with two objects related in multiple ways

Users can still describe this, but the 4- or 9-intersection models are no longer able to represent this.

If one uses the 4- or 9-intersection models, these multiple relations would require a set of matrices, each of which describing one such relation. Furthermore, L touches R three times and also crosses it three times. Again, this cannot be expressed in previous formalizations. We now present a model, named the 3-axis-intersection model, that can be used to express these multiple relations.

Let us now build upon the formalism described in the previous section. We introduce the mechanisms needed to define the binary topological relations in terms of intersections of the topological parts of spatial objects. This new formalism considers the dimension of the intersection. More specifically, we use the 9-intersection matrices for the 0-d, 1-d, and 2-d components as different sets. These components are represented by the three 3 × 3-matrices, $\Im_{3-axis_{0-d}}$, $\Im_{3-axis_{1-d}}$, $\Im_{3-axis_{2-d}}$, which we call the *3-axis-intersections*:

$$\Im_{3-axis_{0-d}}(A, B) = \begin{pmatrix} S_{0,0,0} & S_{0,1,0} & S_{0,2,0} \\ S_{1,0,0} & S_{1,1,0} & S_{1,2,0} \\ S_{2,0,0} & S_{2,1,0} & S_{2,2,0} \end{pmatrix},$$

$$\Im_{3-axis_{1-d}}(A, B) = \begin{pmatrix} S_{0,0,1} & S_{0,1,1} & S_{0,2,1} \\ S_{1,0,1} & S_{1,1,1} & S_{1,2,1} \\ S_{2,0,1} & S_{2,1,1} & S_{2,2,1} \end{pmatrix},$$

$$\Im_{3-axis_{2-d}}(A, B) = \begin{pmatrix} S_{0,0,2} & S_{0,1,2} & S_{0,2,2} \\ S_{1,0,2} & S_{1,1,2} & S_{1,2,2} \\ S_{2,0,2} & S_{2,1,2} & S_{2,2,2} \end{pmatrix}.$$

Each of these three matrices corresponds to nine intersection sets. Each of these usual intersection sets is generated as the union of connected intersection components in a particular dimension (0-d, 1-d, and 2-d), amounting to a total of 27 possible sets $S_{i,j,k}$. The first index (i) indicates the topological part of the first object, the second index (j) specifies the topological part of the second spatial object, while the last index (k) indicates the dimension of the connected components in the set. For example,

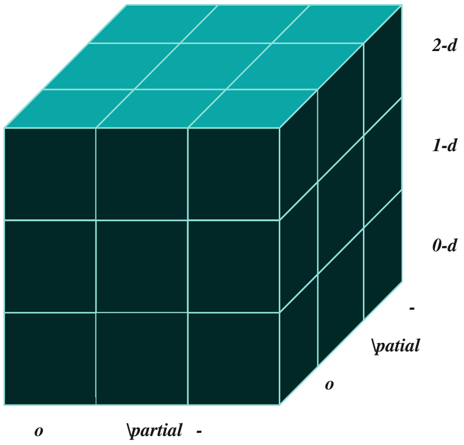


Fig. 3 The 3-axis-intersection model

$S_{1,2,1}$ identifies the intersection set of the boundary of the first object with the exterior of the second object including only 1-dimensional connected components.

The 27 elements can be graphically represented as a 3 × 3 × 3 cube composed of 27 unit cubes, each of which represents an intersection set (see Fig. 3). The cube is depicted in three layers of unit cubes, where the bottom layer corresponds to the 0-*d* matrix, the middle layer corresponds to the 1-*d* matrix, and the topmost layer corresponds to the 2-*d* matrix.

The domain of our formalism is the number of connected components per intersection ($Dom_{3-axis}(S) = \{0, 1, 2, \ldots\}$). Now, instead of just describing the absence of an intersection or its highest dimension, $S_{i,j,k}$ "counts" the number of connected components of an intersection. Figure 2 is now described using this formalism by means of three matrices:

$$\Im_{3-axis_{0-d}}(L, R) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\Im_{3-axis_{1-d}}(L, R) = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 2 & 4 \\ 0 & 6 & 0 \end{pmatrix},$$

$$\Im_{3-axis_{2-d}}(L, R) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 0 & 3 \end{pmatrix}.$$

For instance, $S_{1,1,1} = 2$ denotes that the boundaries of L and R intersect twice in terms of 1-*d* connected components. The 7 intersection sets with non zero values have their geometric interpretations shown in Fig. 4. For instance, consider $\Im_{3-axis_{2-d}}$, which describes the intersections of L and R with respect to 2-*d* connected

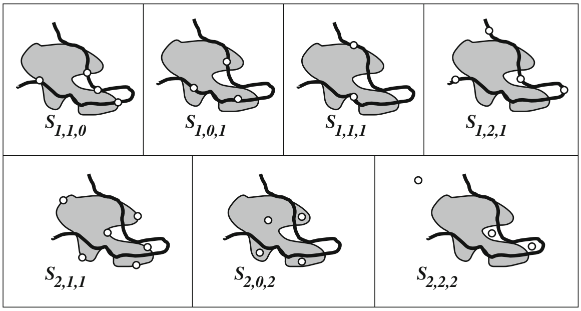


Fig. 4 The geometric interpretations of the set of intersections

components. $S_{2,0,2} = 4$ because there are 4 connected components in the intersection set $(L^{-} \cap R^{\circ})$.

It may appear discouraging that there might be $(n+1)^{27}$ possible relations between two objects, where n is the maximum number of connected components. However, a rather remarkable consequence of our restricting the spatial objects to be connected, is that most such combinations are impossible for objects embedded in the plane due to their topological properties (Egenhofer and Franzosa 1991; Egenhofer and Herring 1991a) and their codimensions (Egenhofer and Herring 1990; Herring 1991; Pigot 1991).

Since the dimension of the intersection cannot be higher than the lowest dimension of the object parts involved, some elements of the matrices are impossible (non-occurring), denoting an unfeasible relation. When a given matrix element is impossible, the corresponding unit cube does not exist. Figure 5 shows the cubes for each relation group after impossible elements are discarded. It shows, for instance, that there are at most 6 (point/point) intersection sets and at most 22 (region/region) possible intersection sets. The point/point and region/region relations determine the lower and upper bounds of the possible relations between two objects. Thus, we have $[(n+1)^6, (n+1)^{22}]$ possible relations, but the number is still unlimited.

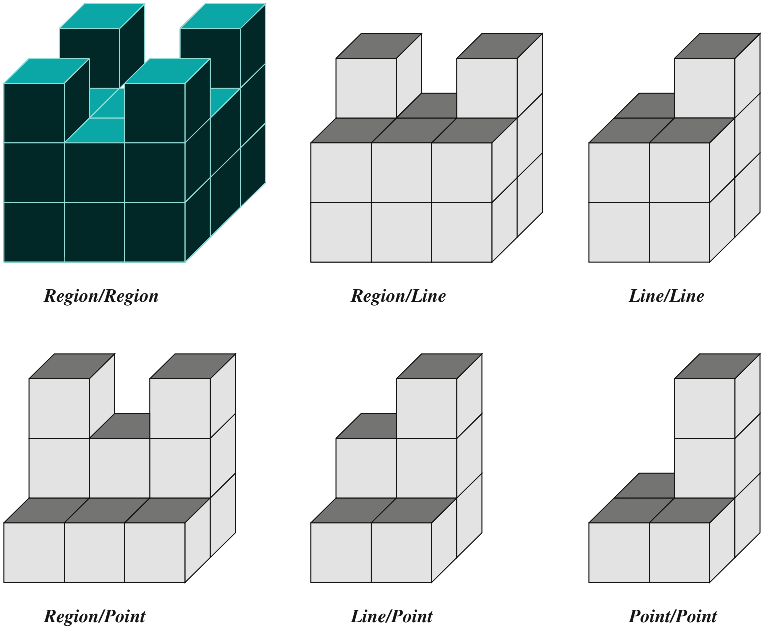
Further simplification comes from the fact that the sets of intersections can only have connected components with the highest possible dimension. The only exception is in the boundary/boundary intersections that can have 0-d and 1-d components. Thus, the lower and upper bounds become $[(n+1)^4, (n+1)^{10}]$ possible relations.

Note that the boundary/boundary intersections ($S_{1,1,0}$ and $S_{1,1,1}$) determine the maximum value, m, in the 3-axis intersection. Let c_1 and c_2 be integers such that

$$Dom_{3-axis}(S_{1,1,0}) = \{0, \ldots, c_1\}$$

and

$$Dom_{3-axis}(S_{1,1,1}) = \{0, \ldots, c_2\}.$$



Region/Region *Region/Line* *Line/Line*

Region/Point *Line/Point* *Point/Point*

Fig. 5 The dimension of the intersection restricting the model

Table 2 The number of relations for all relation groups

Relation groups	3-axis-intersection
Region/region	10
Region/line	35
Line/line	42
Region/point	3
Line/point	5
Point/point	2

By restricting c_1 and c_2 to at most a certain constant value c, we are able to bound m to at most $(2 \times c) + 1$. By employing an appropriate value of c, we can control the maximum value in the 3-axis intersection and limit the number of possible relations to $[((2 \times c) + 2)^4, ((2 \times c) + 2)^{10}]$.

Hereafter, we assume $c = 1$, that is, $Dom_{3-axis}(S_{1,1,0}) = Dom_{3-axis}(S_{1,1,1}) = \{0, 1\}$. The number of feasible relations that can then be expressed is displayed in Table 2. This table shows, for instance, that one can define at most 35 different ways in which a line and a region can relate topologically simultaneously when $c = 1$ (*i.e.*, when there is at most one connected component per intersection).

The generalization of our approach to consider complex spatial objects (*e.g.*, a region composed of disjoint regions possibly including holes) is simple. Basically, the relationships between two complex objects are described in terms of the relations among their individual components. Formally, let O^1 and O^2 be two complex spatial objects $O^1 = \{o_1^1, \ldots, o_{n_1}^1\}$ and $O^2 = \{o_1^2, \ldots, o_{n_2}^2\}$, where o_i^k are the individual components of O^k for $k = 1, 2$ and $i = 1, \ldots, n_k$, and n_k is the number of individual components of O^k. Each individual component of a complex object is a connected simple object. We represent the topological relation $\Im(O^1, O^2)$ between O^1 and O^2 as the set of topological relations $\Im(o_i^1, o_j^2)$ between the simple objects corresponding to pairs (o_i^1, o_j^2) of individual components o_i^1 and o_j^2 of the respective complex objects O^1 and O^2 for $i = 1, \ldots, n_1$ and $j = 1, \ldots, n_2$.

5 Conceptual Neighbourhood Diagram

In order to help the user describe topological relations, they can be organized in a diagram based on the *snapshot model* (Egenhofer and Mark 1995; Freksa 1992). The snapshot model derives the conceptual neighborhoods among topological relations considering their degree of similarity. Conceptual neighborhood diagrams are used to schematize spatio-temporal relations. They enable spatio-temporal reasoning to infer properties of the relations, list possible transitions of a particular relation (*e.g.*, find a sequence of spatial configurations between two relations), and relax query constraints by including neighboring relations.

Conceptual neighborhood diagrams are constructed according to a particular smooth transformation or to a specific similarity measure. In this paper, we consider a topological distance (Egenhofer and Al-Taha 1992) based on the3-axis-intersection model. The *topological distance* $\tau(r_a, r_b)$ between two topological relations r_a and r_b is the sum of absolute values of the differences of corresponding elements in the 3-axis-intersection model:

$$\tau(r_a, r_b) = \sum_{i=0}^{2} \sum_{j=0}^{2} \sum_{k=0}^{2} |S_{i,j,k}^a - S_{i,j,k}^b|.$$

The shortest non-zero distance among all pairs of topological relations determines that two relations are considered conceptual neighbors. They are represented by graphs in which each relation is depicted as a node and conceptual neighbors are linked by edges.

Let us consider the 35 relations between a line and a region shown in Fig. 6. The 35 situations of region/line group can be presented in a conceptual neighborhood diagram. Each relation is a conceptual neighbor of at least one, and at most five other relations. The diagram is disconnected and has one subgraph G_1 with 15 nodes and another G_2 with 20 nodes. This diagram is disconnected because we only consider conceptual neighbors with a small (*i.e.*, close to the minimum) topological distance. This diagram has a particular symmetry with respect to the center, where on the left-hand side are all relations in which some parts of the line are inside the

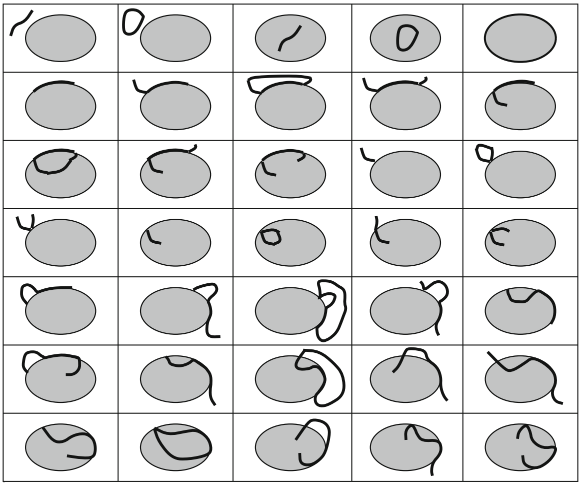


Fig. 6 The 35 region/line topological relations distinguished by the 3-axis-intersection

region, while on the right-hand side are the relations in which the corresponding parts of the line are outside. These diagrams allow partitioning groups of relations into more general cases, which in turn help users express the underlying spatial concepts.

6 Conclusions and Future Work

This paper presented a new framework for modeling topological relations among objects of type point, line, and region, which subsumes and extends previous work. It allows specification of complex scenarios where two objects are spatially related in more than one way. These results allow increasing the flexibility that users are offered to model their reality, thereby contributing to research in formal methods in naive geography.

In terms of expressiveness, our new approach displays a much higher discriminative power such that any relation represented by previous models will also have a unique 3-axis-intersection matrix. For example, even though all points in a line are considered boundary points, our model can differentiate between two open lines which cross at the middle and a line that terminates at its endpoint on another line. In the first situation, we have 5 components resulting from the intersection: the cross point and the four pieces of both lines (2 per line). In the second situation, we have only 4 components: the point where the lines touch, the entire second line (but the touch point), and the two 1-*d* pieces of the first line. The fact that the entire line is considered as a boundary does not decrease the representative power of our model. However, the fact that we count intersection components at each dimension for all 9 intersections, actually, increases the number of possible relations represented using our model. The intersection at end points of lines have a clear implication in the intersection sets of our model: the number of connected components differ as mentioned above. Therefore, the descriptive power of our model does not decrease due to our definition of topological parts (specifically boundary of lines). On the contrary, the fact that we count the number of components at each dimension for all possible 9 intersections increases significantly the number of relations that can be represented. By introducing the counting of components, the number of different relations described with our model is infinite. However, if we bound the maximum number of components per dimension per intersection, we bound the total number of feasible relations.

References

Alboody A, Inglada J, Sedes F (2009) Enriching the spatial reasoning system RCC8. SIGSPATIAL Spec 1(1):14–20

Alexandroff P (1961) Elementary concepts of topology. Dover Publications, Inc., New York

Clementini E, Felice PD (1995) A comparison of methods for representing topological relationships. Inf Sci Appl 3(3):149–178

Clementini E, Felice PD, van Oosterom P (1992) A small set of formal topological relationships suitable for end-user interaction. Technical Report, University of L'Aquilla, Italy

Clementini E, Sharma J, Egenhofer M (1994) Modelling topological spatial relations: strategies for query processing. Comput Graph 18(6):815–822

Deng M, Cheng T, Chen X, Li Z (2007) Multi-level topological relations between spatial regions based upon topological invariants. GeoInformatica 11(2):239–267

Egenhofer M (1991) Extending SQL for cartographic display. Cartogr Geogr Inf Syst 18(4):230–245

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

Egenhofer M, Franzosa R (1991) Point-set topological spatial relations. Int J Geogr Inf Syst 5(2):161–174

Egenhofer M, Herring J (1990) A mathematical framework for the definition of topological relationships. In: Brassel K, Kishimoto H (eds) Proceedings fourth international symposium on spatial data handling, vol 2. International geographical union, Zurich, Switzerland, pp 803–813

Egenhofer M, Herring J (1991) Categorizing binary topological relationships between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Orono

Egenhofer M, Herring J (1991) Categorizing topological spatial relations between point, line, and area objects. Technical Report, University of Maine

Egenhofer M, Mark D (1995) Modelling conceptual neighbourhoods of topological line-region relations. Int J Geogr Inf Syst 9(5):555–565

Frank A (1982) Mapquery–database query language for retrieval of geometric data and its graphical representation. ACM Comput Graph 16(3):199–207

Freksa C (1992) Temporal reasoning based on semi-intervals. Artif Intell 54:199–227

Herring J (1991) The mathematical modeling of spatial and non-spatial information in geographic information systems. In: Mark D, Frank A (eds) Cognitive and linguistic aspects of geographic space. Kluwer Academic Publishers, Dordrecht, pp 313–350

Herring J, Larsen R, Shivakumar J (1988) Extensions to the SQL language to support spatial analysis in a topological database. In: Proceedings GIS/LIS'88, San Antonio, TX, pp 741–750

Kurata Y (2009) Semi-automated derivation of conceptual neighborhood graphs of topological relations. In: Proceedings of the international conference on spatial information theory

Kurata Y, Egenhofer M (2007) The 9^+-intersection for topological relations between a directed line segment and a region. In: Proceedings of the workshop on behavioral monitoring and interpretation, vol 42, pp 62–76

McKenney M, Pauly A, Praing R, Schneider M (2005) Dimension-refined topological predicates. In: Proceedings GIS'05, Bremen, Germany

Papadias D, Egenhofer M, Sharma J (1996) Hierarchical reasoning about direction relations. In: Proccedings 4th ACM workshop on advances in, GIS, pp 107–114

Pigot S (1991) Topological models for 3d spatial information systems. In: Mark D, White D (eds) Proceedings of autocarto 10 (Falls church: american society for photogrammetry and remote sensing, pp 368–392

Raper J, Bundock M (1991) UGIX: A layer-based model for a GIS user interface. In: Mark D, Frank A (eds) Cognitive and linguistic aspects of geographic space. Kluwer Academic Publishers, Dordrecht, pp 449–476

Roussopoulos N, Faloutsos C, Sellis T (1988) An efficient pictorial database system for PSQL. IEEE Trans Software Eng 14(5):630–638

A Time-Driven Mobile Location Aware Service for Distributing Campus Information

Jung-hong Hong, Zeal Su and Steve Yeh

Abstract LBS-based applications heavily rely on location constraints to select information to meet users' demands. Many decisions in real life, however, must also consider the influence of temporal constraints. This research intends to develop a location-aware push service that provides useful messages of campus events to students by taking both spatial and temporal perspectives into consideration. Temporal information is additionally included as an essential type of constraint to determine if a message would be pushed to users. This implies that the offered information not only changes with the movement of users, but also changes as times go by. The prototype of Location-Aware Pushing Service (LAPS) is designed to continuously and automatically sending useful information from the campus database to students according to their changing status. As the number of students using smart phones continuously increases, the proposed mechanism can be further enhanced to enable schools to instantly select and effectively distribute valuable information to enrich students' campus lives on an individual student basis.

Keywords Location-based · Location-aware · Push service

J.-H. Hong (✉) · Z. Su
Department of Geomatics, National Cheng Kung University, Tainan, Taiwan
e-mail: junghong@mail.ncku.edu.tw

Z. Su
e-mail: zeal0820@gmail.com

S. Yeh
GIPS Geographic Technology Co., Ltd, Tainan, Taiwan
e-mail: aapplle123@gmail.com

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_6,

1 Introduction

As the number of mobile devices, especially smart phones and pad devices, rapidly increased in recent years, location-based services (LBS) (Iris et al. 2008) have attracted a tremendous attention from a variety of technology domains. Serving as an effective and powerful communication mechanism to distribute information to potential users, the LBS technology creates a brand new business market to venders of data production, service, hardware, communication and any enterprise that is explicitly or implicitly related to the use of geospatial information. With the booming of wireless communication, users nowadays can easily access the Internet practically at any time and from anywhere, the LBS technology thus already facilitates a "ubiquitous" data distribution and working environment to humans' everyday lives. Compared to the text-based web applications, the LBS-based applications hold a distinct advantage that the response from LBS servers is by its nature "spatially" customized to meet users' application needs. It is therefore no surprise that LBS-based apps have been remaining in the top-ranking list in current app market (e.g., Android and iOS) (Shek 2010). A recent example is an app developed to supply public transportation information for the Taipei city (Taipei City Department of Transportation, 2012). It quickly attracted a tremendous number of downloads as soon as it hit the market. Based on users' location, this app continuously updates the arrival time of the bus they are waiting for. Since users are constantly aware of when the bus is coming from the updated information displayed on the screen, they can comfortably plan their actions without worrying they might miss the bus. This example clearly demonstrates how a well-designed LBS-based app can rapidly and revolutionarily improve the quality of humans' lives. Many similar successful examples using LBS technology can be found in fields like navigation, tourist, exhibition center guidance, etc.

Most of the current mobile applications are developed on the basis of so-called "pull services" (Mohammadi et al. 2011), where the service responds with data only upon users' requests. Due to its passive nature, users are often unaware of the real-time status if they do not continuously issue data request. In contrast, a "push service" is designed to automatically supply information according to users' continuously changing statuses, such that users are aware of things of interests as they move. Since the information is automatically pushed, it is extremely important to avoid repeatedly sending nonsense or annoying information to users. In addition to the locational constraints, we argue that temporal constraints must be also considered in LAS-based applications. A time-driven approach to aid the data selection in LAS-based applications is proposed in this paper. The task of distributing campus information to students is chosen to test the feasibility of the proposed algorithm because schools are operated according to rigorous schedules to create an ordered learning environment for students. The ultimate goal is to successfully send useful information to students to enrich their campus lives. The paper is organized as follows: Sect. 2 reviews related work in LBS research; Sect. 3 discusses our design of the push algorithm; Sect. 4 discusses the development of the prototype system and Sect. 5 demonstrates its test results. Finally, Sect. 6 concludes the major findings and suggests future work.

2 Related Work

Service technology has been extensively used for sharing data with users from other domains (Erl 2005). The concept of a Service-Oriented Architecture (SOA) (Papazoglou et al. 2007) has revolutionized the development of software. From a technological perspective, system developers can easily develop and flexibly extend their applications by composing or chaining services from authorized organizations to reduce unnecessary and duplicated spending. It has been estimated that 80 % of the distributed information explicitly or implicitly include a spatial component (Franklin 1992), so there have been a variety of recent applications using geospatial web service technology to enable the communication between different groups of users. Especially after the OGC proposed common geospatial services standards such as WMS, WFS or WPS (Open Geospatial Consortium 2012), geospatial information can now be readily distributed and interpreted in a transparent and interoperable way. Since the volume of available resource in the Internet steadily increases, the most critical challenge nowadays is no longer how to distribute data. Instead, the intelligence of helping users to select and use the "correct data" that precisely meet their application needs should receive much more attention. The recent progress of intelligent search engines clearly demonstrates how service providers wish to not only take advantage of the available resources, but also to simplify users' processes of decision-making (Oh et al. 2011; Wang and Yin 2011).

Although LBS-based applications have been successfully implemented in a variety of domains, most of them nonetheless are based on locational constraints only. More intelligence on the selection of data and analysis of users' behaviors must be developed. Meier and Cahill (2010) proposed a location-aware, event-based middleware for mobile applications. It supports the event-based programming paradigm and provides an event service well suited for supporting location-aware mobile applications. Context-aware recommender systems (CARSs) (Adomavicius and Tuzhilin 2008) are also proposed to take the mobile user's current or future location into consideration. Results demonstrate that the addition of the contextual information can lead to the generation of more accurate recommendations and the development of more efficient algorithms (Zheng et al. 2009; Fan and Zhi-Mei 2009). In Paelke et al. (2010), the authors provide mobile users with information on the spatial context of a location or a selected route. They also present an approach that can gather context information from freely available sources like Wikipedia. When considering the limitations of mobile applications, the push technology appears to be an effective way to deliver or synchronize commercial information to clients (Lee 2009; Kim et al. 2009). Using push mechanisms initiated by servers, users do not need to navigate the service content and advertisers do not need to passively wait for clients' requests. In Tongyu et al. (2010), an advanced model called "push-pull service" is proposed to include both the advantages of pull and push.

Although there has been a lot of progress in LBS technology concerning how data is distributed and visualized, knowledge about data selection for LAS-based applications is still in great demand. Such applications are characterized by their capability to automatically push useful information to users, but no user would be happy to receive outdated information or events impossible to reach in time, so considering temporal constraints is necessary. A push algorithm that considers both the distance and temporal constraints will be discussed in the following sections.

3 The Design of the Push Algorithm

The information distribution of campus events is selected as the major focus in this paper, so the primary modeling targets are students and events. Our goal is to design an algorithm that can automatically select and push useful information about campus events to students. At this moment, only spatial and temporal constraints are considered, constraints on the themes and context of the events are temporarily excluded and will be added in future research. Both students and events have their corresponding spatial and temporal information. The locations of students may change, while the location of events is normally fixed. After accessing information of students' class timetables from the campus database, the system knows where and when students are (at a particular classroom during the course time) and can thus determine their free time (e.g., time periods between courses). The major idea for the proposed algorithm is to select campus events not only from the distance perspective (e.g., a buffer of 100 m from the current location), but also the temporal perspective (e.g., the time required for moving to a place and participating in the entire event).

3.1 Representation of Time

Time is a mandatory type of constraint in our algorithm. In GIS research, the primitive way for modeling temporal status is either by a point (a time instance) or an interval (a period of time) (Allen 1983). For example, the timestamp of users' location is often modeled as a time instance, while the timestamp for an event is often modeled as a time interval. A time interval t can be presented as (t^-, t^+), where t^-denotes the time the event begins and t^+ denotes the time it ends. For any given event e_i, its corresponding temporal information will be represented by t_i in the following discussion for simplicity.

3.2 Predictable and Unpredictable Mode

Assuming every student has his or her own profile, three types of information can be acquired from the class timetables: the courses he or she takes, the time of the courses and the location of the classrooms of those courses. Although the timetable can only supply information about when and where a student will be during his or

Table 1 The distance score

Mode	Distance score
Predictable (P-Mode)	$\gamma_\mathrm{i} = \varepsilon_\mathrm{i}.\mathrm{o} + \varepsilon_\mathrm{i}.\mathrm{d}$
Unpredictable (U-Mode)	$\gamma_\mathrm{i} = \varepsilon_\mathrm{i}$

her course time, the free time between different courses and how far this student can reach can now be determined. The designed push algorithm will include two modes: predictable (*P-Mode*) and unpredictable mode (*U-Mode*). P-mode is used when students' next scheduled destination is available, for example, the next class begins at 2 p.m. at room 201 of Building 1301. The U-Mode, on the other hand, is used when no such information is available, for example, what students will do after the last class is over. Whether to trigger P-Mode or U-mode can be easily determined by the information acquired from the class timetable.

3.3 Distance Score

The basis for LBS-based applications is to supply information in the nearby area of users' current location. The location of user j and the current time is respectively represented as lj and t_{now}. Based on the users' current location and the location of the next destination (e.g., an event or the next class), the distance between them can be easily calculated using routing algorithm (Zhang and He 2012). Note that the distance may continuously change as users move. In P-Mode, the route distance from l_j to the next destination at t_{now} is represented as ε_j. The symbol E denotes a set of events that may be of interests to users. For any event $e_i \in E$, the location of e_i is represented as $l.e_i$, the distance from l_j to $l.e_i$ is represented as $\varepsilon_i.o$, and the distance from $l.e_i$ to the next destination is represented as $\varepsilon_i.d$. This implies that users will move to the location of the event first ($\varepsilon_i.o$), and then continue to move to the next destination ($\varepsilon_i.d$). In U-Mode, only the distance from l_i to $l.e_i$, represented by ε_i, needs to be considered. To evaluate the results from the perspective of distance, the distance score γ_i for an event e_i in P-mode and U-mode are respectively defined in Table 1.

3.4 Time Score

If the information of students' free time is available, the system shall only push information on events in which students have sufficient time to participate. Because this must consider the time required for both movements and the event, it is necessary to add temporal constraints into the algorithm design. The campus events are subdivided into two major categories: atomic and dividable. Atomic events denote events that require students to participate in the entire event (e.g., concert, movie, speech, etc.), while dividable events denote events that students can freely join or leave (e.g., dining, exhibition, etc.). Similar to the idea of distance score, the "time score" for an event is suggested.

Time Buffers. For atomic events in P-Mode, two types of buffered time for students' movement are considered, one is the time required to move from user's current location to the place of the event, and the other is from the place of the event to the next destination. These two phases of movement are defined as pre-buffer and post-buffer in Fig. 1, where ρ_i and σ_i, respectively represent the time required for the movements. By further combing the time interval T_i of an event e_i, we can determine the total time required to participate in the event e_i. The relationships between these three temporal parameters can be represented by Eqs. (1) and (2):

$$\begin{cases} \rho_i = \frac{\varepsilon_i.\mathrm{o}}{v} \\ \rho_i^+ = T_i^- \end{cases} \tag{1}$$

$$\begin{cases} \sigma = \frac{\varepsilon_i.\mathrm{d}}{v} \\ \sigma_i^- = T_i^+ \end{cases} \tag{2}$$

The parameter v denotes the walking speed of users, which can be assigned by systems based on users' profiles or calculated by the GPS tracking of users. The atomic time interval of participating in the event e_i is defined as $T_i.a$:

$$T_i.a = (\rho_i^-, \sigma_i^+) \tag{3}$$

For atomic events in U-Mode, we only need to consider the pre-buffer for any event e_i, so $T_i.a$ is defined as:

$$T_i.a = (\rho_i^-, T_i^+) \tag{4}$$

For dividable events in P-Mode, both the pre-buffer ρ_i and the post-buffer σ_i must be considered when calculating the total cost for participating in an event e_i. For U-Mode, only the cost of the pre-buffer is considered.

Value of Time Score. The free time of user j is represented as λ_j. Table 2 shows the definition of time score β_i for any event e_i in P-mode and U-mode. In this paper, the value of time score β_i is either 0 or 1, i.e., this parameter only determines if the information of a particular event will be pushed, but does not determine which one is better. For atomic events, the value of β_i depends on there being sufficient time for students to move and participate in the entire event. For dividable events, the value of β_i depends on students' free time allowing them to at least reach the place of the event and continue to move on to the next destination.

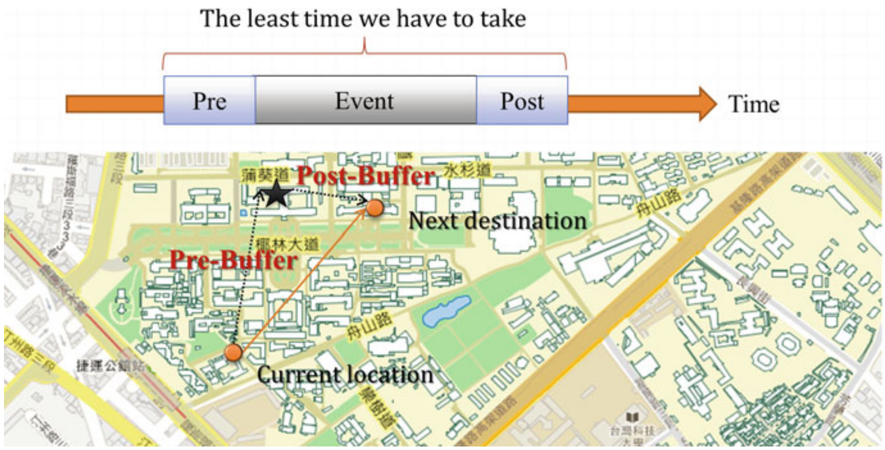


Fig. 1 Pre-buffer and post-buffer in P-Mode

Table 2 The time score

Category	Mode	Time score β_i
Atomic	P-Mode	$\begin{cases} \beta_i = 1 & if\ Ti.a \subseteq \lambda_j \\ \beta_i = 0 & otherwise \end{cases}$
	U-Mode	$\begin{cases} \beta_i = 1 & if\ Ti.a \subseteq \lambda_j \\ \beta_i = 0 & otherwise \end{cases}$
Dividable	P-Mode	$\begin{cases} \beta_i = 1 & if \begin{cases} (t_{now} + \rho_i.o + \rho_i.d) < T_i^+ \\ (t_{now}, t_{now} + \rho_i.o + \rho_i.d) \subseteq \lambda_j \end{cases} \\ \beta_i = 0 & otherwise \end{cases}$
	U-Mode	$\begin{cases} \beta_i = 1 & if \{tnow, tnow + \rho_i.o) \subseteq T_i \\ \beta_i = 0 & otherwise \end{cases}$

3.5 *Timing for Push Service*

In our push service, the push procedure is automatically triggered whenever user location and temporal information qualify the given constraints. In real life, there are two types of events we want to push: positive and negative events. Positive events denote events users have interest in (e.g., food on sale advertisement) or deem useful (e.g., safe routes in the night). For negative events, a repelling push is triggered to warn users that some events may cause inconvenience to their plans (e.g., academic building under construction, suspended classes, etc.). In addition to these two conditions, some types of event information are always pushed whenever they happen (e.g., emergency, fire alarm, etc.).

Under normal circumstances, users would like to be aware of anything that may cause inconvenience, so negative events are automatically pushed. Positive events are pushed if the push score is non-zero and less than the threshold value. The push

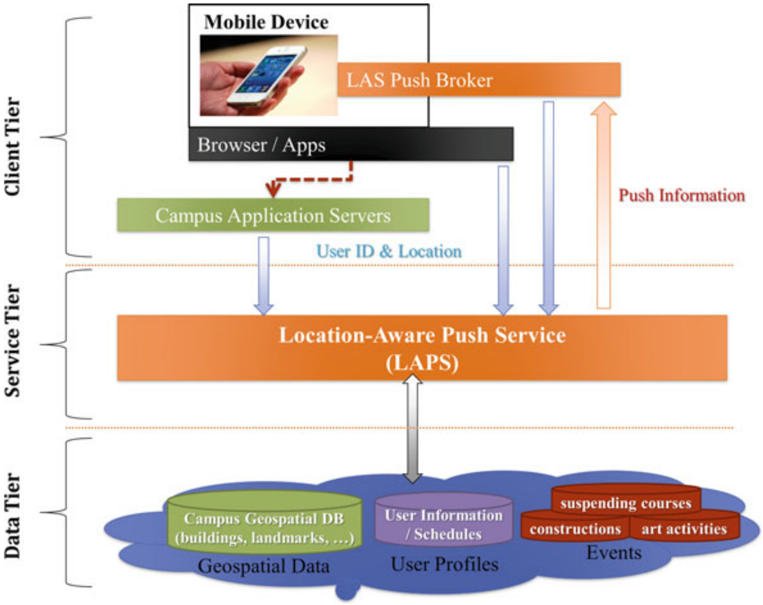


Fig. 2 System architecture for the pushing service

score α_i for event e_i is calculated by Eq. (5). As β_i is either 0 or 1, its value determines whether the information on the event is pushed or not. The distance score, on the other hand, is mainly used for evaluating the cost for adding an event to user's schedule.

$$\begin{cases} \alpha_i = \gamma_i \times \beta_i \\ \gamma_i \in \mathbf{R}^+ \\ \beta_i \in \{0, 1\} \end{cases} \tag{5}$$

4 System Architecture

A prototype system of Location-Aware Pushing Service (LAPS) is developed following the proposed algorithm. Figure 2 shows the three tiers of our system architecture, namely, the data-tier, the client-tier, and the service-tier.

4.1 Data Tier

The data tier is designed to store the required geospatial data, user profile, and campus event information. Geospatial databases are used to store a variety of themes of geospatial features about campus (e.g., buildings, landmarks, and roads). To meet the requirements of geospatial processing, the system is implemented using PostgreSQL with the PostGIS (2012) plugin. PostGIS supports the OpenGIS Simple Features Specification for SQL. User profiles and time-related information (e.g., time about events, students' timetables) are stored in relational databases. These types of information were mainly collected from existing campus databases managed by different offices or departments.

4.2 Client Tier

The purpose of the client tier is twofold. It is a data collector that continuously updates the user's current status, and it is also a data presenter that offers visualized information to users for action reference. A location-aware push broker (LAPB) is developed, which aims to receive the pushed messages and update users' location upon his or her permission. The location of users is therefore provided from three sources: apps on the mobile devices, application servers of mobile apps, and our LAPB client.

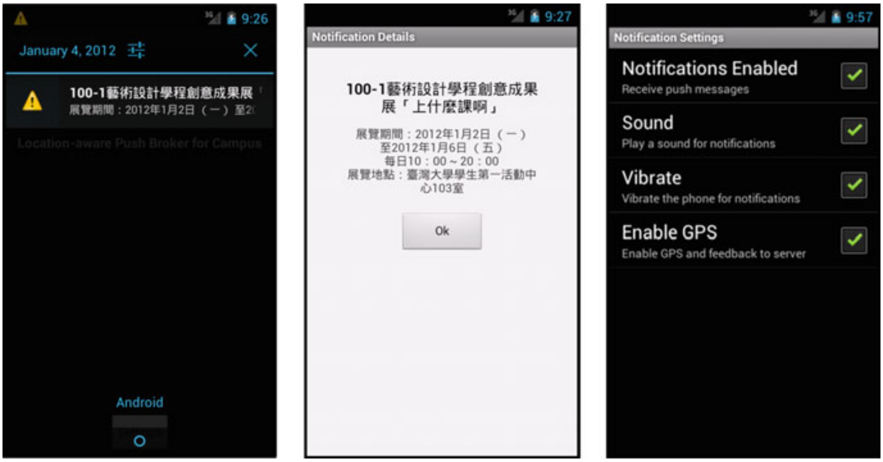
The LAPB is implemented on Android devices using Android SDK v4.0 (Fig. 3). The geolocation API in Android SDK was adopted to provide the location service. We use the Java-based Android SDK to implement the client applications by integrating a number of open source JAVA APIs. The LAPB also uses the Android Push Notification API (Android Push Notification 2011), an open source project for push notifications on the Android platform, as the basis for developing push services. Rooted in the asmack project (Asmack 2010), this API implements the Extensible Messaging and Presence Protocol (XMPP) (XMPP 2010) client using JAVA.

4.3 Service Tier

The major functions of the service tier are to receive the location information from the client tier, to access existing databases and dynamically determine what information shall be pushed to the users. To meet the requirements of flexibility, extensibility and easier integration, the LAPS is developed following the concept of a service-oriented architecture, as shown in Fig. 4.

Service Component. The message exchanges between LAPB and LAPS are based on XMPP, an open-standard communications protocol for message-oriented middleware based on XML. XMPP-based software is widely deployed across the Internet, for example, Google Talk and the chat feature to third-party applications on Facebook. The XMPP server in LAPB is based on Openfire (2012), is written in Java and dual-licensed under both a proprietary license and the Apache License 2.0.

For mobile clients and application servers, we use the RESTful API with JSON messages because REST has received widespread acceptance across the web


9:26

January 4, 2012

100-1藝術設計學程創意成果展
展覽期間：2012年1月2日（一）至2

Location-aware Push Broker for Campus

Android

9:27

Notification Details

100-1藝術設計學程創意成果展「上什麼課啊」

展覽期間：2012年1月2日（一）
至2012年1月6日（五）
每日10：00 ~ 20：00
展覽地點：臺灣大學學生第一活動中心103室

Ok

9:57

Notification Settings

Notifications Enabled
Receive push messages ☑

Sound
Play a sound for notifications ☑

Vibrate
Vibrate the phone for notifications ☑

Enable GPS
Enable GPS and feedback to server ☑

Fig. 3 The implementation of LAPB, the push message in the notification bar, and the notification settings

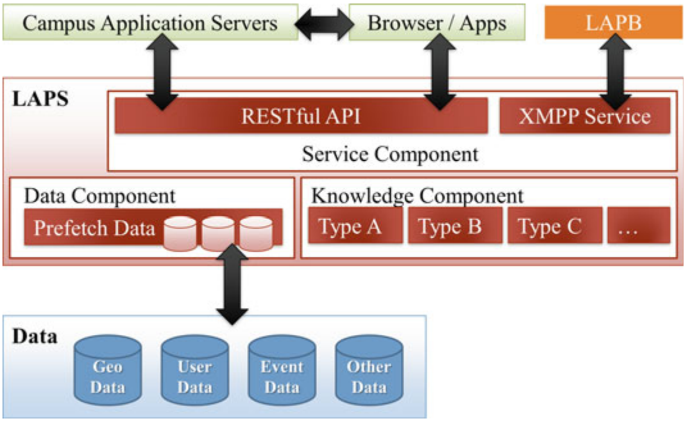


Fig. 4 Service components in LAPS

community as a simple alternative to SOAP with WSDL-based services. The RESTful API provides common services for the development of location-aware applications, for example, User Registry, Event Registry, and Location Feedback.

Data Component and Knowledge Component. LAPB collects data from the data tier and pre-fetch data from campus databases when necessary (e.g., create the index for event databases). Meanwhile, the data collected from the client tier will be saved in the database, too (e.g., registration data and location logs). According to the parameters of the above algorithm, the built-in rule is used to determine if event information will be pushed to the users.

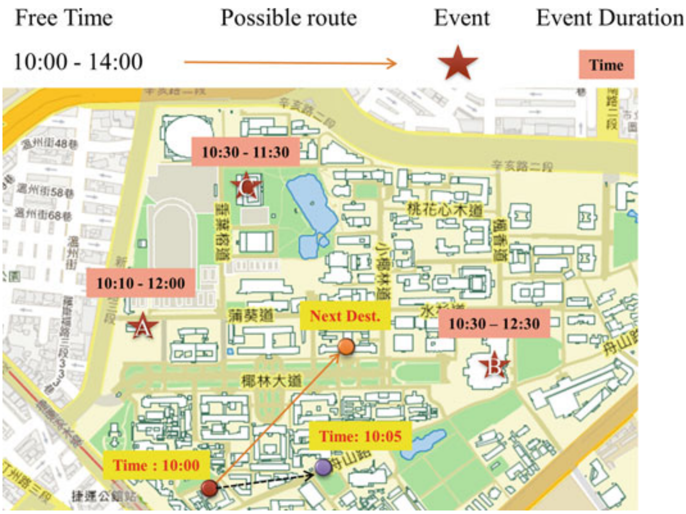


Fig. 5 Location of a user with timestamps

Major Services. LAPS provides a common service interface for location-aware applications on campus. The implemented LAPS can provide information about campus events, such as speeches, workshops, campus activities, emergency situation, road constructions, etc. Threshold values are given to serve as criteria for selecting events.

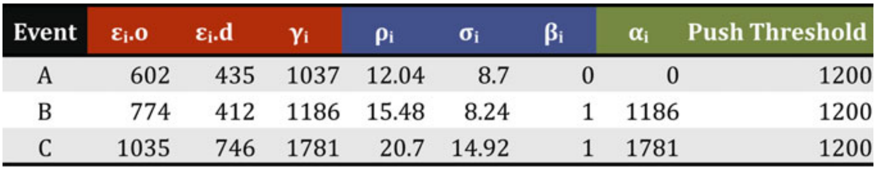
5 Test Analysis

The LAPB is implemented on Android-based mobile devices and LAPS is developed using JAVA-based Technologies. The design of LAPB allows the location information of users to be continuously updated to LAPS (i.e. tracking the location of all users who have installed LAPB on their mobile device). LAPS is responsible for identifying the users and issuing search constraints to the campus databases.

A test example is shown in Fig. 5 In this case, the free time for a user is from 10:00 to 14:00 according to the class timetable. Based on the spatial and temporal constraints, the LAPS automatically filters the candidate events from the existing databases. Three atomic events, A, B and C, are selected according to the user's current location.

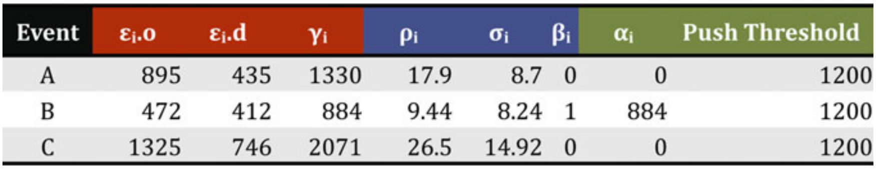
Table 3 shows the results of each parameter based on the proposed push algorithm when the time is 10:00. Despite the fact that event A clearly has the lowest distance score γ_i, its time score β_i is 0, meaning that the information of event A will not be pushed to the users. This is because the pre-buffer time (the time required for the user to move to the place of the event) overlaps with temporal information of the event, which implies that users will not be able to arrive in time. Although both event B and

Table 3 Parameters at the time of 10:00



Event	$\varepsilon_i.o$	$\varepsilon_i.d$	γ_i	ρ_i	σ_i	β_i	α_i	Push Threshold
A	602	435	1037	12.04	8.7	0	0	1200
B	774	412	1186	15.48	8.24	1	1186	1200
C	1035	746	1781	20.7	14.92	1	1781	1200

Table 4 Parameters at the time of 10:05



Event	$\varepsilon_i.o$	$\varepsilon_i.d$	γ_i	ρ_i	σ_i	β_i	α_i	Push Threshold
A	895	435	1330	17.9	8.7	0	0	1200
B	472	412	884	9.44	8.24	1	884	1200
C	1325	746	2071	26.5	14.92	0	0	1200

event C have positive scores, only the score of event B is lower than the threshold value, so only event B is pushed to the user.

Assuming a student selects event B after receiving the pushed message, Table 4 shows the parameter values when the time reaches 10:05. Because both the time and user's location change, the push scores of the three events may also change. The push score of Event C is now zero and will be removed from the candidate list because users will never arrive there in time. It shows that the proposed distance score and time score are time-dependent and the threshold values can be adjusted to adapt to the needs of real applications.

6 Conclusions and Future Work

Many activities and events in LBS applications have explicit temporal components. In addition to the location constraints, a location-aware service must be able to analyze and determine the pushed service content according to the temporal status of the users and events. We proposed an algorithm for selecting pushed service content by taking both factors of distance and time into consideration. Using campus information as an example, the analyzed test shows that such a LAS-based application can improve the quality of the service content and avoid pushing nonsense information. Although the current result is still preliminary and some of the parameters may need to be further adjusted, the advantages of additionally including temporal information in the LAS-based applications are extremely obvious. More improvements can still be added to enhance the performance of the proposed LAPS in the future. For example, the addition of an interest analysis model according to students' historical behaviors can serve as criteria to precisely suggest things users are interested in. The threshold values of time score equations can be further adjusted to adapt to different application scenarios. More types of environmental information, e.g., weather and road closure can also be added to improve the performance. The addition of more parameters and data will make the algorithm more complicated, but it will be able to provide precise suggestions that best fit the users' application preferences in future LAS-based applications.

References

Adomavicius G, Tuzhilin A (2008) Context-aware recommender systems. In: ACM RecSys Tutorial

Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843

Android push notification (2011). http://sourceforge.net/projects/androidpn/

Asmack (2010). http://code.google.com/p/asmack/

Erl T (2005) Service-oriented architecture: concepts, technology, and design. Prentice Hall, Upper Saddle River

Fan Y, Zhi-Mei W (2009) A mobile location-based information recommendation system based on GPS and WEB2.0 services. WSEAS Trans Comput 8(4):725–734

Franklin C (1992) An introduction to geographic information systems: linking maps to databases. In: Database

Iris AJ, Junglas IA, Watson RT, Richard TW (2008) Location-based services, Commun ACM 51(3):65–69

Kim YS, Lee JW, Park SR, Choi BC (2009) Mobile advertisement system using data push scheduling based on user preference. In: Wireless telecommunications symposium, 2009, pp 1–5

Lee JW, Lee CS, Park YS (2009) Research on the advertisement effect of push type mobile advertisement. In: 4th international conference on cooperation and promotion of information resources in science and technology, 2009, pp 137–142

Meier R, Cahill V (2010) On event-based middleware for location-aware mobile applications. IEEE Trans Softw Eng 36(3):409–430

Mohammadi M, Molaei M, Naserasadi A (2011) A survey on location based services and positioning techniques. Int J Comput Appl 24:5

Oh J, Meneguzzi F, Sycara K (2011) Probabilistic plan recognition for intelligent information assistants. In: ICAART

Open Geospatial Consortium (2012). http://www.opengeospatial.org/

Openfire (2012). http://www.igniterealtime.org/projects/openfire/

Paelke V, Dahinden T, Eggert D, Mondzech J (2010) Location based context awareness through tag-cloud visualizations. In: Joint international conference on theory, data handling and modelling in geospatial information science, Hong Kong, 2010, pp 290–295

Papazoglou MP, Traverso P, Dustdar S, Leymann F (2007) Service-oriented computing: state of the art and research challenges. IEEE Comput 40:2007

PostGIS (2010). http://postgis.refractions.net/

Shek S (2010) Next-generation location-based services for mobile devices. CSC grants. Available from http://assets1.csc.com/lef/downloads/Accessed:2012/04/10

Taipei City Department of Transportation (2012). http://www.dot.taipei.gov.tw/

Tongyu Z, Yuan Z, Fei W, Weifeng L (2010) A location-based push service architecture with clustering method. In: Sixth international conference on networked computing and advanced information management, NCM, 2010, pp 107–112

Wang Y, Yin J (2011) Enhanced decision making free search. In: Proceedings of world conference on science and engineering. In: Elsevier

XMPP (2010). http://xmpp.org/

Zhang L, He X (2012) Route search base on pgRouting. Advances in intelligent and soft computing. In: Wu Y (ed) Advances in intelligent and soft computing. Software Engineering and Knowledge Engineering, vol 2. AISC 115, pp 1003–1007

Zheng Y, Zhang L, Xie X, Ma W (2009) Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the WWW' 09, pp 791–800

Evaluating the Cognitive Adequacy of the DLine-Region Calculus

Alexander Klippel, Jinlong Yang, Rui Li and Jan Oliver Wallgrün

Abstract Qualitative spatio-temporal calculi play a crucial role in modeling, representing, and reasoning about geospatial dynamics such as the movement of agents or geographic entities. They are ubiquitous in ontological modeling, information retrieval, they play a central part at the human-machine interface, and are critical to process data collected from geosensor networks. What is common to all these application areas is the search for a mechanism to transform data into knowledge borrowing heavily from strategies of (human) cognitive information processing. Astonishingly, there is paucity in actual behavioral evaluations on whether the suggested calculi are indeed cognitively adequate. While the assumption seems to be made that qualitative equals cognitive, a more differentiated view is needed. This paper is filling the void by the first (to the best of our knowledge) behavioral assessment of the DLine-Region calculus using actual dynamic stimuli. These assessments are crucial as the few experiments that exist have clearly demonstrated that topological relations form conceptual groups (clusters), a fact that seems to be highly likely for the 26 DLine-Region relations as well. Our results show which topological relations form (cognitive) conceptual clusters.

This research is funded through the National Science Foundations (#0924534).

A. Klippel (✉) · J. Yang · R. Li · J. O. Wallgrün
Department of Geography, GeoVISTA Center, Pennsylvania State University,
University Park, PA, USA
e-mail: klippel@psu.edu

J. Yang
e-mail: jinlong@psu.edu

R. Li
e-mail: rui.li@psu.edu

J. O. Wallgrün
e-mail: wallgrun@psu.edu

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_7,

1 Introduction

The meaningful processing of spatio-temporal data is a challenging and recent research topic. The tremendous amount of data that is becoming increasingly available has spurred multidisciplinary efforts to process and analyze data telling various stories of the dynamic earth (e.g., Adrienko et al. 2008; Allen 1983). One important development are qualitative spatio-temporal representation and reasoning (short: QSTR) approaches (e.g., Kurata and Egenhofer 2009; Muller 2002; Sridhar et al. 2011; van de Weghe et al. 2008). The reason for the popularity of qualitative formalisms can be summarized by a quote from Galton's (2000) seminal book on qualitative change:

> The divisions of qualitative space correspond to salient discontinuities in our apprehension of quantitative space.

The importance of this statement is difficult to overestimate. If this statement is true, then we have found—by using QSTR—a way to bridge the gap between cognitive (semantic) information processing and requirements of formal systems fundamental to modern information technologies. From our usage of the word "if" the reader may derive that it is, unfortunately, not that easy. There are numerous suggestions for QSTR that all identify—sometimes contradictory—divisions of quantitative space. Additionally, while often a claim is made that qualitative approaches bear some inherent cognitive/commonsense aspects of how spatio-temporal information is processed, there is a paucity of actual behavioral evaluations that would back up the claim of capturing cognitive aspects of spatio-temporal information processing using QSTR. Hence, we do need an experimental framework that allows for effectively and efficiently assessing QSTR approaches cognitively. This article details an experiment assessing the DLine-Region calculus (Kurata 2008) while at the same time showing general ways of assessing the cognitive adequacy of QSTR approaches that are built on jointly exhaustive and mutually exclusive relations.

The remainder of this article is structured as follows: first, we provide some background information on existing behavioral studies as well as the DLine-Region calculus. Second, we report on the conducted experiment evaluating the divisions (topological equivalence classes) that are inherent to the DLine-Region calculus. The results show that we need to superimpose a hierarchy onto the 26 primitive relations to reflect human conceptualizations of movement patterns captured by this calculus. We discuss in the outlook some strategies how behavioral results can be transformed into weights for conceptual neighborhood graphs (CNGs) that will cognitively adjust their use in areas such as information retrieval, ontology engineering, or the geo-spatial semantic web.

2 Background

Given the space limitations we will focus on some essential behavioral studies and a brief introduction to the DLine-Region calculus. Of particular importance to the topic of this article are the experiments by Mark and Egenhofer (e.g., Mark and Egenhofer 1994a, 1995; Shariff et al. 1998) on topologically characterized relations between a line and a region. In their experiments, the line had no direction and they used static images (rather than animations, see Sect. 3). They focused on both cognitive conceptualization processes as well as the spatial semantics of linguistic expressions. In some of their experiments they employed a grouping paradigm (Mark and Egenhofer 1994b), similar to the approach taken in the experiment we will present later on. Their findings crystalized in the famous statement that *topology matters and metric refines* (Egenhofer and Mark 1995). Not as widely discussed but equally important is their finding that the 19 relations between a line and a region are cognitively not primitive relations. In other words, the 19 relations form conceptual groups (clusters) with larger within group similarity as well as larger between group dissimilarity. This aspect has also been addressed in formal research papers and we will come back to this aspect throughout the paper.

Behavioral research on topology and qualitative calculi addressing actual movement patterns is rare. There are only a few studies that use dynamic stimuli in their experiments. Noteworthy is the research by Lu et al. (2009) in which the authors evaluate Allen's (1983) temporal interval calculus. The results, in a nutshell, indicate that certain relations can be considered forming clusters, mimicking the findings of Mark and Egenhofer (1994a). Specifically, *before* and *meet* form one cluster and all other relations of Allen's calculus form a second one. Our own research (see Klippel et al. 2012 for an overview) so far has addressed topological movement patterns that can be modeled as changing relations between two spatially extended entities. Evaluating RCC-8 (Randell et al. 1992) and Egenhofer's intersection models (Egenhofer and Franzosa 1991) that can be used to characterize these movement patterns, we found that topological relations form conceptual groups (clusters) and that domain semantics is an important contextual factor (Klippel 2012). To the best of our knowledge, there are no other published behavioral evaluations of dynamic movement patterns characterized by qualitative calculi.[1]

Back to the DLine-Region calculus; Kurata and Egenhofer (2007) extended the original 9-intersection models proposed by Egenhofer and Franzosa (1991) such that the direction of a line (in relation to a region) can be captured, too. The original 9-intersection model details the relation between two spatial objects A and B by creating a 3 × 3 matrix that details, from a point-set topological perspective, the relation (intersection) between three topological parts of A and B: interior, boundary, and exterior. For the case of a directed line, Kurata and Egenhofer introduced a finer distinction of the boundary of the line separately representing the start and the end of

[1] There are numerous proposals on general cognitive aspects of spatial dynamics (e.g., Mennis et al. 2000), cognitive studies on events and movement (Shipley and Zacks 2008) and there are also some unpublished studies.

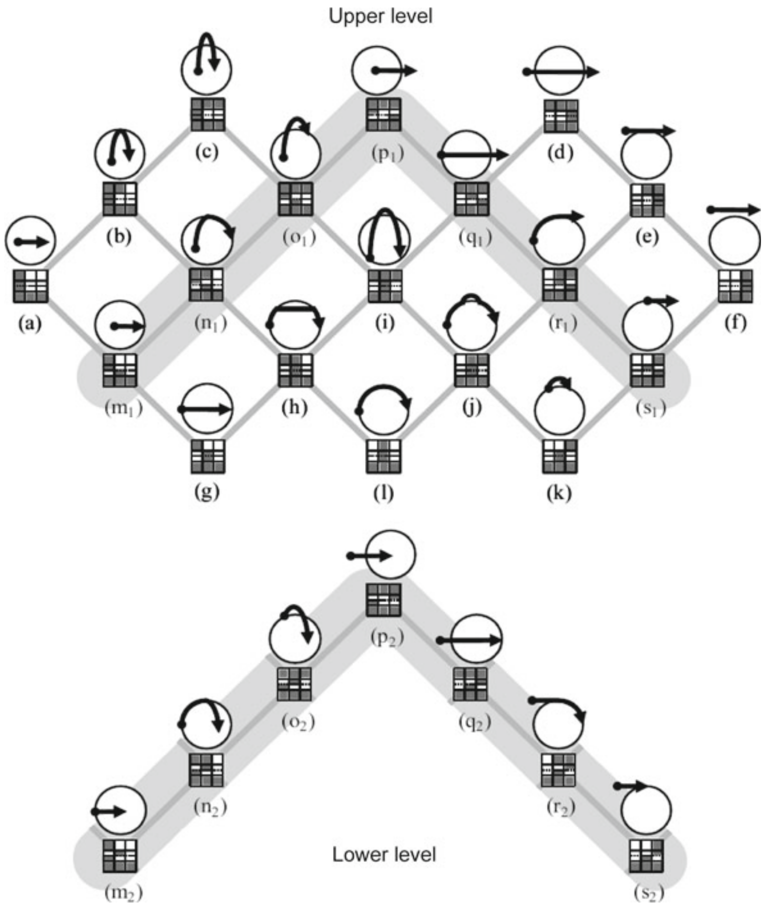


Fig. 1 Conceptual neighborhood graph for the 26 DLine-Region relations (see Kurata and Egenhofer 2009)

a line. This finer distinction results in an extra row in the 9-intersection matrix which then, consequently, is referred to as the 9+-Intersection Model (Kurata 2008). In 2D, 19 relations are possible between a line and a region but there are 26 topologically equivalent relations possible in case the line is directed. Visually these relations can be organized into a conceptual neighborhood graph (see Fig. 1, see also Egenhofer and Al-Taha 1992; Freksa 1992; Randell and Cohn 1989). Two relations, R_1 and R_2, are conceptual neighbors if it is possible for to hold R_1 over a tuple of objects at a certain point in time, and for to hold R_2 over the tuple at a later time, with no other (third) mutually exclusive relation holding in between (Cohn 2008; Freksa 1992). A neighborhood graph has one node for each relation $R \in \mathbf{R}$, and an edge between two nodes if the corresponding relations are neighbors. The important aspect to keep in mind, which will tremendously add to the transformative nature of this paper, is that virtually every calculus with jointly exhaustive and pairwise disjoint (JEPD) relations (such as RCC and the intersection models) has a conceptual neighborhood graph (Cohn and Renz 2008), and that the methods applied here will be universally applicable amongst these calculi to improve their cognitive adequacy.

3 Experiment

We have extended an experimental framework (Knauff et al. 1997; Mark and Egenhofer 1994b) that is tailored to requirements in the spatial sciences in that it allows for evaluating qualitative spatial calculi built on JEPD relations. We are using a grouping paradigm, which is classically employed to reveal cognitive conceptualizations, and combining it with animations based on topological equivalence classes. The equivalence classes for the purpose of this article are the ones specified in the DLine-Region calculus, which we briefly introduced in the previous section. The critical questions that this experiment is answering are:

- Are the equivalence classes identified by the DLine-Region calculus cognitively salient classes of equivalence in our apprehension of (qualitative) space?
- On the basis of previous research it is fair to assume that not all 26 relations are equally salient but that they will form groups. If so, which relations are considered more similar to each other than others?

Participants. 26 Penn State undergraduate students participated for course credit. Average age was 22 (9 female).

Design and materials. 78 randomized animated icons depicting 26 DLine-Region relations (three per equivalence class) were constructed in Adobe Flash CS4. Each icon was 120 by 120 pixels in size. Given that participants see actual animations, we restricted the movement in that it, generally speaking, only went from left to right. Figure 2 shows an example of one DLine-Region relation (d, see Fig. 1). A random start point was selected in the exterior of the start region. The movement of the black dot starts from the left exterior, fully crosses the interior, and then ends at a randomly selected point in the right exterior (the end region). To give another example, for relation (m_1) the dot starts from a randomized location in the interior of the start region, moves in the interior, and ends somewhere on the boundary in the end region.

In the construction of all icons, particular attention was paid to two specific aspects: (1) that the starting and ending relations were perceptually clear; and (2) that the speed of the dot was constant in and among all icons. The speed of the dot movement was kept constant by maintaining the same ratio between the path length and the number of frames. At the end of each dot movement, the dot paused to represent the ending relation before the movement was repeated. The construction of icons involved manual inspection by experts to remove and replace icons that were ambiguous due to the randomized locations of start- and end-points. The final 78 icons did not convey DLine-Region relations that are ambiguous.

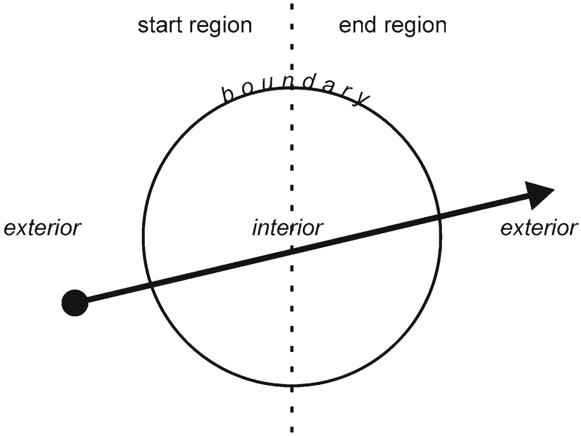


Fig. 2 Icon shows the case of DLine-Region relation (d), where the *dot* starts from a randomized location in the exterior (*left*), fully crosses the interior, and then ends at a randomized location in the exterior (*right*)

Procedure. The experiment took place in a GIS lab and was set up as a group experiment. The lab seats up to 16 participants at the same time at Dell workstations (Optiplex 755, 24" widescreen LCD monitors). View blocks ensured that participants performed this task individually. Participants performed a free classification (category construction) task as well as a linguistic labeling task.

Our custom-made grouping software, CatScan, allows for presenting dynamic stimuli and administers the complete experiment (see Fig. 3). All 78 animations showing the 26 DLine-Region relations were initially displayed on the left side of the screen in random order. The right side was empty and participants were required to create all groups. Animated icons can be placed into groups by simple drag-and-drop operations; they can be placed into groups, out of groups, or moved between groups. In case a group is deleted, all icons are placed back on the left side. The main grouping task was preceded by a warm up grouping task (sorting animals) to acquaint participants with the interface, the grouping environment, and general idea of a free classification task. After finishing the task (no time limit was given), participants were presented with the groups they created and provided linguistic labels for these groups: a short label of no more than 5 words (e.g., *inside out in*) and a longer description (e.g., *found the dots that started inside the circle then went out then back in*).

4 Results

On average, participants created 11.3 groups (min: 3; max 39, statistically an outlier) and it took them 22 min on average to finish the grouping task. The average number of groups is, as expected, below the 26 formally defined topological equivalence

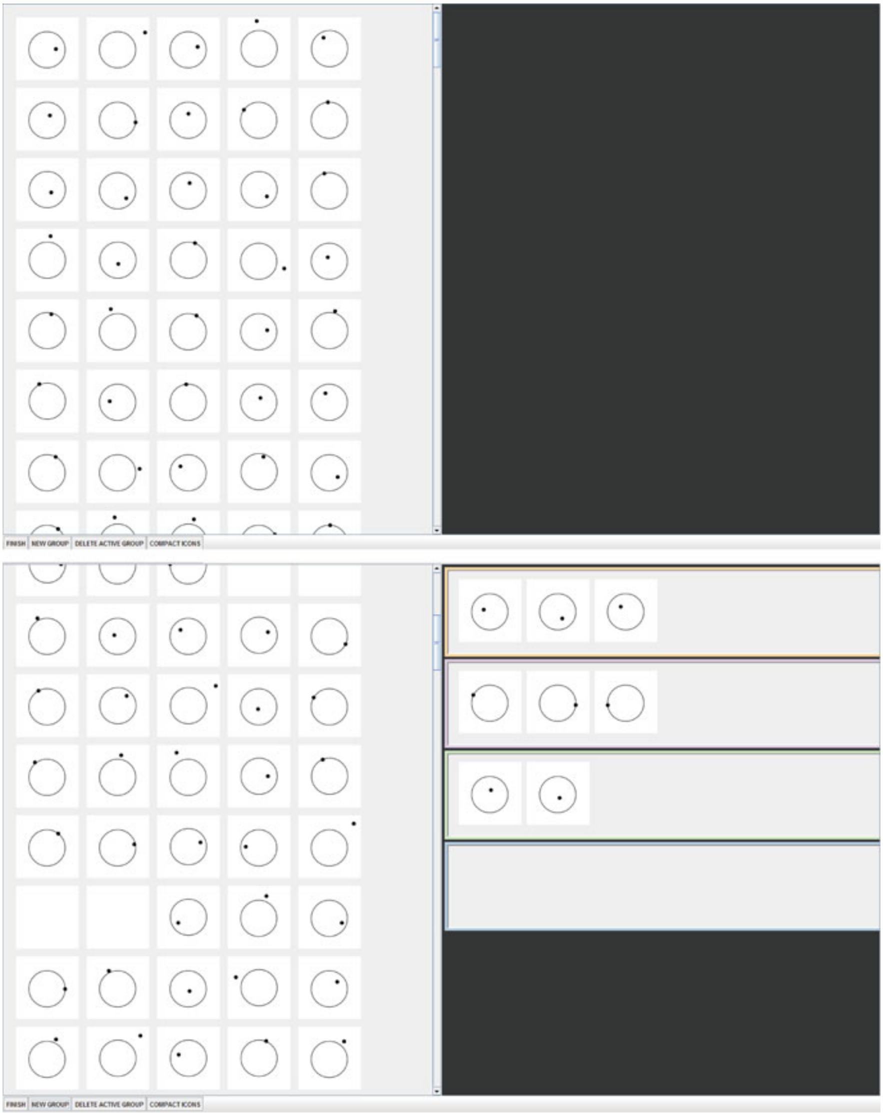


Fig. 3 The screenshots of the experiment interface. The *top* screenshot shows the initial screen before participants started to group. The *bottom* screenshot shows an ongoing experiment with groups created by the participant

classes. This result is in accordance with several findings (Klippel et al. 2012; Mark and Egenhofer 1994b) that topological equivalence classes defined by a number of calculi are forming conceptual clusters, whose formally defined granularity is often not the granularity identified by human participants.

The grouping behavior of participants itself is captured in individual similarity matrices which encode two icons being placed into the same group as 1 and two icons not being placed into the same group as 0. By summing over all individual similarity matrices, an overall similarity matrix (OSM) can be created. Thereby maximum similarity is assigned to those pairs of animations that are always placed together into the same group and minimum similarity is assigned to those pairs of animations that are never placed into the same group. The maximum similarity corresponds to the number of participants (here: $N = 26$), the minimum similarity is 0. This data can be analyzed using a number of techniques such as cluster analysis, multidimensional scaling, or simple heat maps that allow for visualizing raw similarities.

Figure 4 shows a combination of a cluster analysis (Ward's method) and a heat map. Additional annotations in Fig. 4 (dashed boxes) show a synthesis of comparing different cluster methods as a means to validate clustering results (Kos and Psenicka 2000). The two dendrograms (top and left part) are identical and show the result of the cluster analysis; icon names are placed on the bottom and right side of the heat map and consist of the relation name plus a number from 1 to 3. The heat map visualizes raw similarities such that higher similarities are displayed in darker gray and lower similarities are displayed in lighter gray. The combination of heat map and cluster analysis allows for a better interpretation of the grouping behavior as the clustering structure revealed by the dendrograms can be directly related to actual grouping behavior.

A first observation is that topological equivalence is a strong predictor for the similarity ratings; this is indicated by the fact that all three instances of all topological relations are in neighboring columns/rows, that is, they are grouped together most frequently compared to all other grouping possibilities (the dark gray cells along the diagonal, top-left to bottom-right). To this end, these results are largely consistent with data from other experiments (Klippel et al. 2012; Mark and Egenhofer 1994b) that topology at the base level is a strong grouping criterion. Equally interesting is, however, the overall grouping structure, and the combination of heat map and dendrograms allows for a deeper interpretation.

Figure 4 reveals—on a coarse level—a three cluster solution using Ward's method that has, however, to be somewhat modified if three clustering methods (Ward's, average linkage, complete linkage) are compared. The result of this synthesis is indicated by the dashed-line-boxes. To additionally visualize the solution indicated by the combined analysis (i.e. the dashed-line-boxes) we have also depicted these results in Fig. 5 using a merged CNG (compare Fig. 1) and using dashed lines reflecting the cluster solution shown in Fig. 4.

The synthesis of all three clustering methods allows for identifying two large clusters that are identical across all three clustering methods and one cluster that requires refinement. The two large clusters identical across methods are: cluster 1 with DLine-Region relations a, b, m_1, m_2, n_1, n_2, g, h, and l; cluster 2 with DLine-Region relations c, o_1, o_2, i, j, and k; and cluster 3, which has to be analyzed in more detail, contains on the coarsest granularity the following relations: p_1, p_2, q_1, q_2, r_1, r_2, s_1, s_2, d, e, f.

Besides the fact that topological equivalence is at the core of the similarity ratings, it is worthwhile to note that the clusters in general also reflect topologically

induced similarities. In other words, the general clustering structure does not violate topological similarity in the sense that the groups formed by participants are connected subgraphs in the DLine-Region CNG (see Figs. 1 and 5). However, the granularity required by a formal topological characterization does not seem to be reflected in the grouping behavior of the participants. Participants focused on more abstract characteristics of the movement patterns. In the following we summarize these characteristics:
Cluster 1:

- No part of the movement is taking place outside the region, meaning that the intersection of any component of the DLine with the exterior of the region is empty.
- The movement starts either on the border of the region or inside the region.
- The movement ends either inside or on the border.
- Parts of the movement can take place along the border.

Cluster 2:

- Part of the movement has to take place outside the region but it is neither the beginning nor the ending.
- Start and end of the movement can be on the border or in the interior of the region.

Cluster 3:

- Part of the movement has to take place outside the region.
- The movement can end in all three possible locations: outside, inside, and on the border.
- Part of the movement can take place in the interior of the region or not.

As indicated above, clusters 1 and 2 are stable across different clustering methods while cluster 3 seems to require a finer level of analysis, that is, it does not show up consistently across different methods. The clustering structure that is indicated in Figs. 4 (dashed boxed) and 5 (dashed lines) reflects this finer level of granularity at which all three clustering methods (Ward's, average linkage, complete linkage) agree. Topologically this finer level of distinction makes sense in that it reveals connected subgraphs of the CNG that are singled out. This finer level of granularity in cluster 3 can be summarized as follows: 3a contains the relations p_1, q_1, and d; the main characteristic here is that the movement ends outside the region while the starting location can be outside, inside, or on the boundary. Cluster 3b contains relations p_2 and q_2; the main characteristic is that the movement starts outside and ends either inside or on the boundary (after having been inside). Cluster 3c contains relations s_2 and r_2; the movement starts outside and ends on the border without intersection the interior. Cluster 3d (r_1 and s_1) contains relations whose movement starts on the border and ends outside the region without intersecting the interior. Cluster 3e singles out relation f which is the only movement pattern that takes place completely outside the region. Cluster 3f singles out relation e which almost takes place completely outside the region except for the interior of the line that intersects with the boundary of the region.

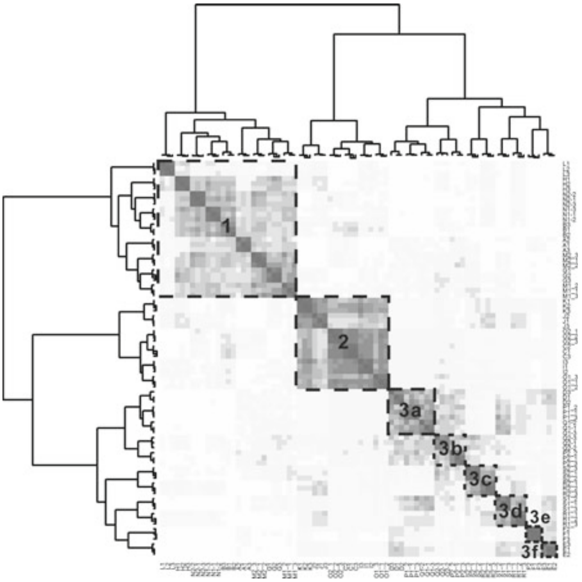


Fig. 4 The figure shows a combination of a heat map and cluster analysis (Ward's method). Additionally, *rectangles* indicate a synthesis of analyzing different clustering methods. The numbers identify clusters discussed in the text. A color and high resolution version of this figure is available at min.us/mSDH2012_figure4

Finally, we would like to point out the somewhat special role of relation l. While it does seem to be integrated into cluster 1, the analysis of the raw similarities clearly indicates its different role in this cluster. This is not surprising as movement pattern l takes place completely on the boundary of the region.

The cluster analysis shows obvious that we need a coarser granularity of movement primitives than offered by the DLine-Region calculus. As revealed by the general characteristics of the three clusters, important distinctions are made based on the movement in relation to the region focusing particularly on distinction of inside versus outside movements or combinations thereof. We therefore looked into the linguistic descriptions (short labels) that participants provided. As the grouping behavior differs from participant to participant we looked into the descriptions here from a general level using a word cloud to reveal frequencies of individual terms (additionally we performed an actual quantitative word count). Figure 6 shows the results that seem to reinforce our interpretation that the three basic distinctions made in the intersection models (including the 9+ model) are crucial, that is, movements are distinguished on whether they are going in or out, and whether they take place inside or outside. The most frequently used movement related terms are (in this order): *in*, *out*, *to*, *on*, *outside*, *straight*, *inside*, *through*, and *arch*.

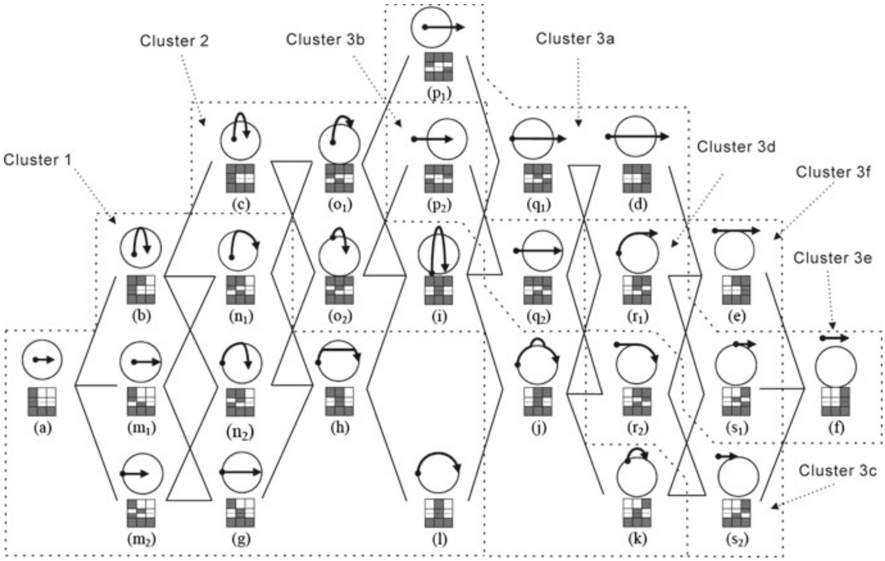


Fig. 5 Merged DLine-Region CNG reflecting the analysis discussed in Fig. 4, Sect. 4. The color coding corresponds to the clusters identified by comparing three clustering methods. A color and high resolution version of this figure is available at min.us/mSDH2012_figure5

Interestingly, most of these terms can be directly related to some topological characteristics. However, there are also several non-topological terms such as *straight* and *arch*. We will discuss the latter in the next section.

5 Discussion

In our discussion of the results, we will draw comparisons to the previously mentioned research by Mark and Egenhofer (1994b, referred to henceforth as E&M94). Although their research setup focused on line-region relations and methodologically used a different set-up, several of their findings are worth looking into in the light of our results.

One of E&M94's central findings indicated that the most frequently grouped icons are those belonging to the same equivalence class (identified by the 9-intersection model). We do find similar results in our data visualized in Fig. 4 indicated by the fact that all three instances of all topological relations are in neighboring columns/rows, that is, they are grouped together most frequently compared to all other grouping possibilities. This shows that topology is indeed a major grouping criterion at the fine granularity analysis level.

Equally important, E&M94 found a comparable pattern with respect to creating clusters of line-region relations, that is, cluster analysis reveals that the 26 original

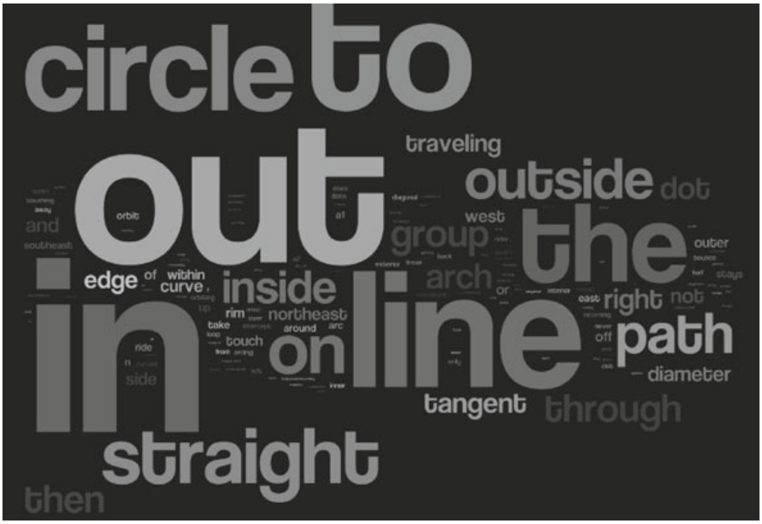


Fig. 6 Wordle showing the frequencies (larger meaning more often used) of words used by participants to characterize their groups (short labels)

DLine-Region relations are conceptually not the primitives. Additionally, we found similarities in that larger clusters are formed by relations on the left side of the CNG (compare Fig. 10 in E&M94 and Fig. 5). While there are an additional 7 relations in the DLine-Region calculus and several differences in the research methodology, this is an interesting finding. In both CNGs the left side has movement patterns that do not take place outside the region while the right side shows more variety including the exterior of the region as a place where movement happens. Taking additionally into consideration the linguistic analysis (see Fig. 6), we find that in/out and inside/outside distinctions are both linguistically and conceptually a dominating criterion.

E&M94's analysis also singled out several relations as most likely forming their own clusters. Specifically, relations a, l, and f (for English speaking participants). We have discussed already the somewhat prominent situations of relations l and f, that is, that in our experiments they form more individualistic clusters as well. That relation a is not singled out may have to do with the different experimental setup: According to current event conceptualization theories, the ending relations of movement patterns play an important role in their conceptualization (Regier and Zheng 2007). From this perspective relation l is not that different from, for example, relation m_2. However, E&M94's results are slightly different for different language groups (English vs. German). Our research focused on English speaking participants but we will discuss the opportunity to use our research framework in cross-linguistic studies in the outlook.

As the linguistic analysis reveals, participants did not group solely on the basis of topological information although it may appear this way. The question of the primacy of topology as a way to think spatially has received more attention recently

(Klippel et al. 2012, Schwering A (2011) Does metric really define topology, personal communication, May 20th, 2011) calling into question the seminal statement that “topology matters and metric refines” (Egenhofer and Mark 1995). Our experiment was not explicitly designed to address this question. However, there are certain constraints that inevitably lead to the introduction of factors other than topology to realize the 26 DLine-Region relations. For example, we used a circle as our region and thereby making it impossible for a straight line to start and end in the circle while having been outside it in between. Most prominently this aspect is featured in linguistic descriptions *straight* and *arch*, which made it into the top ten of the most frequent terms. There is current research in event conceptualization that stresses the importance of path characteristics on the conceptualization of movement patterns (Maguire et al. 2011; Shipley and Maguire 2008).

6 Conclusions

To conclude, our research adds to the body of knowledge that asserts that distinctions—in form of equivalence classes—made by qualitative formal calculi are not necessarily the ones that are foundational to the human cognitive system. Several research approaches have shown that the granularity of formal calculi is inadequate for modeling human conceptualizations of both static and dynamic spatial relations. To address this issue the majority of approaches (e.g., Clementini et al. 1993; Schneider and Behr 2006) define formal criteria on how to cluster topological equivalence classes such that the overall number of topological predicates is reduced. Clementini et al. (1993) suggest five basic relations that are derived on the basis of the emptiness and non-emptiness of component intersection, inclusion and non-inclusion of one object in another object, and the dimension of the component intersection. Schneider and Behr (2006) developed a method based exclusively on the emptiness and non-emptiness of component intersection. Coming from a database user perspective, Schneider and Behr developed the concepts of *topological cluster predicates* and *topological predicate groups* to reduce the number of predicates in a user-defined or application-specific manner.

Our research adds to this body of knowledge by having behaviorally evaluated the DLine-Region calculus that requires, formally, the distinction of 26 relations between a directed line and a region based on an extended version of Egenhofer’s intersection models. Kurata and Egenhofer (2009) discuss several approaches on reducing the 26 relations as their primary goal is to model human concepts of motion. While we find some commonalities in the approaches they discuss and the results of our experiments, there is no complete agreement between any of the discussed approaches and our results. This demonstrates once more the importance of behavioral evaluations of qualitative calculi that is often called for (e.g., Clementini et al. 1993) but rarely delivered.

Given the ubiquity of qualitative spatio-temporal calculi as tools and means to bridge the gap between formal systems and human conceptions of space and time, our research has the potential to provide insights into necessary adaptations/modifications of qualitative calculi to deserve the label “cognitively adequate”. Our research methodology, in general, is tailored to calculi based on jointly exhaustive and pairwise disjoint relations that all form conceptual neighborhood graphs (Cohn and Renz 2008). The use of JEPD and CNGs in areas such as linguistics (Ross et al. 2010), robotics, database query languages and information retrieval, and ontological modeling (e.g. for the semantic web) can be greatly enhanced by behavioral research that provides the necessary bridge between cognitive and formal spatio-temporal semantics.

Future research directions are manifold given both the ubiquity of QSTR in research and application and the paucity of behavioral evaluations. The following ones strike us as important: we have recently demonstrated that domain semantics has a meaningful influence on the grouping behavior of participants, that is, which original topological relations form cognitive conceptual clusters (Klippel 2012). To this end, a theory is needed that would allow for specifying meta-domain characteristics and how they influence cognitive conceptualizations of movement patterns.

Mark and Egenhofer (1994b) raised already the question of the influence of language on the conceptualization of line-region relations. Linguistic (and potentially cultural) influences surface in the spatial science sporadically (see Mark et al. 2007 for a more substantial treatment) but are still not integrated into core theories. A transdisciplinary research agenda is needed to deliver results that could influence spatial theories more fundamentally.

We have developed an approach to characterize movement patterns based on the notion of conceptual primitives (Klippel 2011) that is built on the basic distinctions used to define topological relations between a line and a region: interior, exterior, and boundary (the latter distinguished as movement on a spot and extended movement on the boundary, see also Kurata and Egenhofer 2009). This approach is comparable to work on movement patterns by Stewart Hornsby and Cole (2007) in that basic topological distinctions constitute primitive distinctions that could be additionally annotated by using, for example, direction information. We consider it important to deepen this line of research as it potentially allows for linking linguistic expressions with formal spatial characterizations more flexibly and has the capability of modeling and interpreting continuous movement behavior.

References

Adrienko G, Adrienko N, Dykes J, Fabrikant SI, Wachowicz M (2008) Geovisualization of dynamics, movement and change: key issues and developing approaches in visualization research. Inf Vis 7(3/4):173–180

Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843

Clementini E, Di Felice P, van Oosterom P (1993) A small set of formal topological relationships suitable for end-user interaction. In: Abel D, Ooi BC (eds) Advances in spatial databases. Third international symposium, SSD ’93 Singapore, 23–25 June 1993: proceedings. Springer, Berlin, pp 277–295

Cohn AG (2008) Conceptual neighborhood. In: Shekhar S, Xiong H (eds) Encyclopedia of GIS. Springer, Boston, p 123

Cohn AG, Renz J (2008) Qualitative spatial representation and reasoning. In: van Harmelen F, Lifschitz V, Porter B (eds) Foundations of artificial intelligence. Handbook of knowledge representation, 1st edn. Elsevier, Amsterdam, pp 551–596

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

Egenhofer MJ, Franzosa RD (1991) Point-set topological spatial relations. Int J Geogr Inf Syst 5(2):161–174

Egenhofer MJ, Mark DM (1995) Naive geography. In: Frank AU, Kuhn W (eds) Proceedings of spatial information theory. A theoretical basis for GIS. International conference, COSIT 95, Semmering, Austria, 21–23 Sept 1995. Springer, Berlin, pp 1–15

Freksa C (1992) Temporal reasoning based on semi-intervals. Artif Intell 54(1):199–227

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

Klippel A (2012) Spatial information theory meets spatial thinking—is topology the Rosetta Stone of spatio-temporal cognition? Ann Assoc Am Geogr (in press)

Klippel A (2011) Movement choremes: bridging cognitive understanding and formal characterization of movement patterns. Top Cogn Sci 3(4):722–740

Klippel A, Li R, Yang J, Hardisty F, Xu S (2012) The Egenhofer-Cohn hypothesis: or, topological relativity? In: Raubal M, Frank AU, Mark DM (eds) Cognitive and linguistic aspects of geographic space—new perspectives on geographic information research, Springer, Berlin (in press)

Knauff M, Rauh R, Renz J (1997) A cognitive assessment of topological spatial relations: results from an empirical investigation. In: Hirtle SC, Frank AU (eds) Spatial information theory: a theoretical basis for GIS. Springer, Berlin, p 206

Kos AJ, Psenicka C (2000) Measuring cluster similarity across methods. Psychol Rep 86:858–862

Kurata Y (2008) The 9+-intersection: a universal framework for modeling topological relations. In: Cova TJ, Miller HJ, Beard K, Frank AU, Goodchild MF (eds) Geographic information science. 5th international conference, GIScience 2008, Park City, UT, USA, 23–26 Sept 2008: proceedings. Springer, Berlin, pp 181–198

Kurata Y, Egenhofer MJ (2007) The 9+-intersection for topological relations between a directed line segment and a region. In: Gottfried B (ed) TZI technical report, 1st workshop on behaviour monitoring and interpretation (BMI'07), in conjunction with 30th German conference on artificial intelligence, vol 42. Universität Bremen, pp 62–76

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

Lu S, Harter D, Graesser AC (2009) An empirical and computational investigation of perceiving and remembering event temporal relations. Cogn Sci 33:345–373

Maguire MJ, Brumberg J, Ennis M, Shipley TF (2011) Similarities in object and event segmentation: a geometric approach to event path segmentation. Sp Cogn Comput 3:254–279

Mark DM, Egenhofer MJ (1994a) Calibrating the meanings of spatial predicates from natural language: line-region relations. In Waugh TC, Healey RG (eds) Advances in GIS research, 6th international symposium on spatial data handling, pp 538–553

Mark DM, Egenhofer MJ (1994b) Modeling spatial relations between lines and regions: combining formal mathematical models and human subject testing. Cartogr Geogr Inf Syst 21(3):195–212

Mark DM, Egenhofer MJ (1995) 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

Mark DM, Turk AG, Stea D (2007) Progress on Yindjibarndi ethnophysiography. In: Winter S, Kuipers B, Duckham M, Kulik L (eds) Spatial information theory. 9th international conference, COSIT 2007, Melbourne, Australia, 19–23 Sept 2007: proceedings. Springer, Berlin, pp 1–19

Mennis J, Peuquet DJ, Qian L (2000) A conceptual framework for incorporating cognitive principles into geographical database representation. Int J Geogr Inf Sci 14(6):501–520

Muller P (2002) Topological spatio-temporal reasoning and representation. Comput Intell 18(3):420–450

Randell DA, Cui Z, Cohn AG (1992) A spatial logic based on regions and connections. In: Nebel B, Rich C, Swartout WR (eds) Proceedings of the 3rd international conference on knowledge representation and reasoning. Morgan Kaufmann, pp 165–176

Randell D, Cohn A (1989) Modelling topological and metrical properties in physical processes. In: Brachman R, Levesque H, Reiter R (eds) Proceedings of the1st international conference on the principles of knowledge representation and reasoning. Morgan Kaufmann, Los Altos, pp 55–66

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

Ross RJ, Hois J, Kelleher J (eds) (2010) Computational models of spatial language interpretation (CoSLI) workshop at spatial cognition 2010: CEUR workshop proceedings

Schneider M, Behr T (2006) Topological relationships between complex spatial objects. ACM Trans Database Syst 31(1):31–81

Shariff AR, Egenhofer MJ, Mark DM (1998) Natural-language spatial relations between linear and areal objects: the topology and metric of English-language terms. Int J Geogr Inf Sci 12(3):215–246

Shipley TF, Maguire MJ (2008) Geometric information for event segmentation. In: Shipley TF, Zacks JM (eds) Understanding events: how humans see, represent, and act on events. Oxford University Press, New York, p 435

Shipley TF, Zacks MJ (eds) (2008) Understanding events: how humans see, represent, and act on events. Oxford University Press, New York

Sridhar M, Cohn A, Hogg D (2011) From video to RCC8: exploiting a distance based semantics to stabilise the interpretation of mereotopological relations: spatial information theory. In: Egenhofer M, Giudice N, Moratz R, Worboys M (eds) Lecture notes in computer science. Spatial information theory. 10th international conference, COSIT 2011, Belfast, ME, USA, 12–16 Sept 2011: proceeding. Springer, pp 110–125

Stewart Hornsby K, Cole S (2007) Modeling moving geospatial objects from an event-based perspective. Trans GIS 11(4):555–573

van de Weghe N, Billen R, Kuijpers B, Bogaert P (eds) (2008) Moving objects: from natural to formal language: workshop held in conjunction with GIScience 2008, Utah

Yuan M, Hornsby KS (2008) Computation and visualization for the understanding dynamics in geographic domains: a research agenda. CRC Press, Boca Raton

Integrating Change Vector Analysis, Post-Classification Comparison, and Object-Oriented Image Analysis for Land Use and Land Cover Change Detection Using RADARSAT-2 Polarimetric SAR Images

Zhixin Qi and Anthony Gar-On Yeh

Abstract This study proposes a new method for land use and land cover (LULC) change detection using RADARSAT-2 polarimetric SAR (PolSAR) images. The proposed method combines change vector analysis (CVA) and post-classification analysis (PCC) to detect LULC changes using RADARSAT-2 PolSAR images based on object-oriented image analysis. A hierarchical segmentation was implemented on two RADARSAT-2 PolSAR images acquired at different times to delineate image objects. CVA was applied to the coherency matrix of PolSAR images to identify changed objects, and then PCC was used to determine the type of changes. The classification of the RADARSAT-2 images is based on the integration of polarimetric decomposition, object-oriented image analysis, decision tree algorithms, and support vector machines (SVMs). In comparison with the PCC that is based on the Wishart supervised classification, the proposed method improves the overall error rate for change detection and the overall accuracy for change type determination by 25.15 and 6.59 % respectively. The results show that the proposed method can achieve much higher accuracy for LULC change detection using RADARSAT-2 PolSAR images than the PCC that is based on the Wishart supervised classification.

Keywords Land use and land cover (LULC) · Change detection · RADARSAT-2 · Polarimetric SAR (PolSAR) · Change vector analysis (CVA) · Post-classification comparison (PCC)

Z. Qi (✉) · A. G.-O. Yeh
Department of Urban Planning and Design, The University of Hong Kong,
Pokfulam Road, Hong Kong SAR, People's Republic of China
e-mail: qizhixin@hku.hk

A. G.-O. Yeh
e-mail: hdxugoy@hku.hk

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_8,

1 Introduction

Timely LULC change information is essential for urban planning and management. Remote sensing data obtained from different optical sensors have been commonly used to characterize and quantify LULC information. However, conventional optical remote sensing is limited by weather conditions. There are difficulties in collecting timely LULC information in regions frequently covered by clouds. Radar remote sensing, which is not affected by cloud conditions, is therefore an effective tool for extracting timely LULC change information in those regions. Compared with conventional single-polarization SAR, polarimetric SAR allows for the discrimination of different types of scattering mechanisms that leads to a significant improvement in the quality of classification results. Therefore, PolSAR data has more potential than single-polarization SAR data for LULC change detection.

Numerous methods for change detection that use remote sensing data have been developed in past studies. Reviews of existing change detection methods can be found in many papers (Singh 1989; Coppin et al. 2004; Lu et al. 2004). The current change detection methods can be summarized into two categories: the unsupervised approach and the supervised approach. Among the most widely used unsupervised approaches are image differencing (Weismiller et al. 1977), image rationing (Howarth and Wickware 1981), and change vector analysis (Lambin and Strahler 1994). These methods perform change detection by directly comparing images acquired at different times. The unsupervised approach is relatively simple, straightforward, and easy to implement and interpret. However, this approach cannot provide information on the type of LULC changes. Whereas, the supervised approach can provide information on both changed areas and the types of change these areas undergo because this approach implements change detection based on separate supervised classification of multi-temporal images. Post-classification comparison (PCC) is a widely used supervised method for change detection. However, the accuracy of PCC is limited by the accuracy of the classification. The change result exhibits accuracies similar to the product of the accuracies of each individual classification (Stow et al. 1980). When applied to SAR images, PCC yields poor accuracy because the accuracy of the classification of SAR images is not as high as that of optical images due to the speckle effect and limited information of sing-frequency SAR data. The hybrid change detection method combines the advantages of the unsupervised and the supervised approach has a potential for LULC change detection using PolSAR images. The unsupervised approach can be used to detect changed areas and then PCC is used to classify the changed areas to determine the types of change. Therefore, the hybrid method can effectively reduce the impact of each individual classification on the change detection result and then provide information on the types of change. The combination of the unsupervised and the supervised approach was carried out in change detection using optical remote sensing data (Petit and Lambin 2001; Silapaswan et al. 2001). However, studies on the application of the hybrid method in change detection using PolSAR images are still limited.

The objective of this study is to develop a new method for LULC change detection using RADARSAT-2 PolSAR images. The proposed method combined CVA and PCC to detect LULC changes using RADARSAT-2 PolSAR images based on object-oriented image analysis. A hierarchical segmentation technique was applied on RADARSAT-2 images acquired at different times to delineate image objects. CVA was implemented on the coherency matrix of PolSAR images to identify changed objects, and then PCC was used to determine the type of changes. The classification of PolSAR images was implemented based on the integration of polarimetric decomposition, object-oriented image analysis, decision tree algorithms, and support vector machines (SVMs). First, different polarimetric decomposition techniques were used to extract polarimetric parameters that were then combined with the elements of the backscattering and coherency matrix to form a multichannel image. Second, the object-oriented image analysis was utilized to extract various textural and spatial features to support classification. Third, the decision tree algorithm was used to select features used for classification. Finally, the PolSAR images were classified using SVMs based on the selected features.

2 Study Area and Data

The study area is located in the Panyu District with latitudes 22°53′26″N to 22°59′1″N and longitudes 113°30′31″E to 113°39′57″E of Guangzhou City in southern China. Panyu lies at the heart of the Pearl River Delta, and has a total land area of 1,314 km^2 and a population of 926,542. This district was an agricultural country before the economic reform in 1978 but has been transformed recently into a rapidly urbanized area. Since Panyu became a district of Guangzhou in July 2000, intensive land development has occurred to provide housing to residents of Guangzhou City. Huge profits have been generated through property development, which results in the increase of land speculation activities and illegal land developments. Accurate and timely LULC change information is important for the local government to make management policies to control and prevent illegal land developments at an early stage. Two RADARSAT-2 Fine Quad-Pol images (Single Look Complex) acquired on March 21, 2009 and September 29, 2009 were used to extract LULC change information (Fig. 1). The images have a full polarization of HH, HV, VH, and VV, a resolution of 5.2 × 7.6 m, and an incidence angle of 31.5°.

LULC classes in the study area can be summarized into seven categories, which are urban/built-up (UB), water (W), barren/sparsely vegetated land (BS), forest (F), lawn (L), banana (B), and cropland/natural vegetation (CN). Field investigations were carried out simultaneously with the acquisition of the images to collect ground truth. In the investigations, field plots were selected across the typical LULC classes using a clustered sampling approach (Congalton and Green 2009). A GPS was used to record the coordinates of these field plots. On the basis of the experience with multinomial distribution (Congalton and Green 2009), we collected a minimum of 50 samples for each category. Based on the field plots collected in the field investigations, a total of 1,990 training and validation objects were selected for the classification of

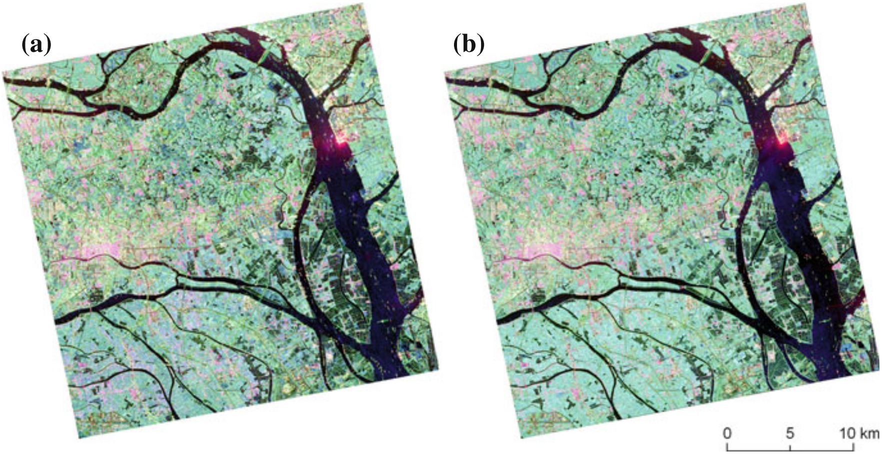


Fig. 1 RADARSAT-2 PolSAR images (Pauli RGB composition): **a** image acquired on March 21, 2009; **b** image acquired on September 29, 2009

Table 1 Number of the plots and pixels selected for each LULC class

Class	Training		Validation	
	Objects	Pixels	Objects	Pixels
Banana (B)	120	54,982	113	42,529
Barren/sparsely vegetated land (BS)	119	22,365	109	24,114
Forest (F)	139	48,861	118	36,437
Lawn (L)	123	44,723	98	34,159
Cropland/natural vegetation (CN)	164	82,865	176	84,670
Urban/built-up areas (UB)	213	60,809	224	65,939
Water (W)	144	64,427	130	66,130
Total	1,022	379,032	968	353,978

PolSAR images. The training group had 1,022 objects, whereas the validation group had 968 objects. The first group was used to select features for classification, and then the second group was used to verify the results of the classification. The plots and pixels selected for each LULC class in the training and validation groups are shown in Table 1. The validation samples were also used as no-change samples to evaluate change detection results. A total of 374 changed image objects were selected as samples of change through visual interpretation and field investigations. Visual interpretation was conducted to identify changed areas from the entire images. The selected changed objects were then validated in field investigations. The number of plots of different types was determined in the light of actual change. More samples were selected for types with more changes, whereas fewer samples were selected for types with fewer changes. The plots and pixels of each change type are shown in Table 2.

Table 2 Number of the plots and pixels selected for different change types

Change type	Plots	Pixels
Banana to water	8	2,280
Barren/sparsely vegetated land to cropland/natural vegetation	74	32,129
Barren/sparsely vegetated land to urban/built-up areas	51	14,341
Barren/sparsely vegetated land to water	48	10,951
Lawn to cropland/natural vegetation	18	7,348
Cropland/natural vegetation to banana	5	3,905
Cropland/natural vegetation to barren/sparsely vegetated land	13	4,125
Cropland/natural vegetation to urban/built-up areas	2	538
Cropland/natural vegetation to water	78	17,934
Urban/built-up areas to barren/sparsely vegetated land	9	1,717
Urban/built-up areas to cropland/natural vegetation	1	40
Water to barren/sparsely vegetated land	11	3,934
Water to lawn	7	4,089
Water to cropland/natural vegetation	49	19,454
Total	374	122,785

3 Methodology

3.1 Preprocessing of RADARSAT-2 PolSAR Images

Image preprocessing included radiometric correction, speckle filtering, and image registration. Radiometric calibration of the RADARSAT-2 images was performed using PolSARPro_v4.1.5 software and applying the sigma look-up table provided in the product. After radiometric correction, the pixel values of the images could be directly related to the radar backscatter of the scene. This is necessary for the comparison of PolSAR images acquired at different times. A Lee Sigma filter with a window size of 7 × 7 was applied on the images to reduce speckles. Compared with other commonly used filters, this one effectively retains subtle details and preserves the shape of small land parcels while reducing the speckle effect (Lee et al. 2009). The advantage of this filter is allowing for the accurate delineation of tiny land parcels in object-oriented image analysis. Image registration was based on the geometric rectification of the RADARSAT-2 images. PCI Geomatica software was used to implement the geometric rectification of the images. The RADARSAT-2 image package provides a total of 180 tie points evenly distributed across the entire image. These tie points tie the line/pixel positions in image coordinates to geographical latitude/longitude and can be used as ground control points (GCP) to register an image to a geocoded target image. This work first created a blank geocoded image with the same resolution as the RADARSAT-2 images and then registered the two RADARSAT-2 images to this geocoded image using PCI Geomatica based on the tie points. Visual inspection indicates that these two images were registered perfectly.

3.2 Object-Oriented Image Analysis for Change Detection

Speckles in PolSAR images significantly affect LULC change detection and classification (Qi et al. 2012). LULC classification based on pixels usually yields poor results with high spatial heterogeneity due to the presence of speckles. Fragmental false alarms could occur in the result of change detection if the detection is conducted on the pixel level. Object-oriented image analysis can be used to reduce the effect of speckles by implementing change detection and classification based on image objects (Qi et al. 2012). Furthermore, additional textural and spatial features extracted from image objects can help improve the accuracy of the classification of PolSAR data (Qi et al. 2012).

A straightforward approach for delineating image objects is to segment images acquired at different times separately and then overlay them together. However, this method will produce inconsistencies in the delineation of boundaries of objects and result in a large number of fragmental patches in the final segmented image. Such excessive fragmentation can lead to difficulties in change detection. A hierarchical image segmentation procedure was proposed to minimize the inconsistency in delineating objects from multi-temporal PolSAR images. In considering two co-registered images, image (t_1) and image (t_2), acquired over the same area at different times t_1 and t_2, the procedure of the hierarchical segmentation can be summarized as follows:

- The initial segmentation is applied to image (t_1) with a fixed scale parameter.
- The same segmentation process is implemented again on image (t_2) while the segmentation result of image (t_1) is taken as the thematic layer for constraint. This procedure would cause all object-merging to take place within the boundaries of the segmentation of image (t_1). New objects will only be created in places where the two images are significantly different.

The hierarchical segmentation technique can eliminate inconsistencies in delineating image objects from multi-temporal PolSAR images. New objects created from the segmentation of image (t_2) and those that do not exist in the segmentation of image (t_1) are changed objects. As shown in Fig. 2, the image object highlighted in yellow shows that changed areas can be delineated using the hierarchical segmentation technique. Although the hierarchical segmentation technique can be used to drive changed objects, changed objects are not just the new objects created in the second segmentation. The hierarchical segmentation only creates new objects if parts of an image object in the first segmentation change, but it will not create any object if the entire object changes. The image object with red color in Fig. 2 shows this situation. Therefore, additional change detection techniques should be used to identify changed objects from all the objects created through the hierarchical segmentation.

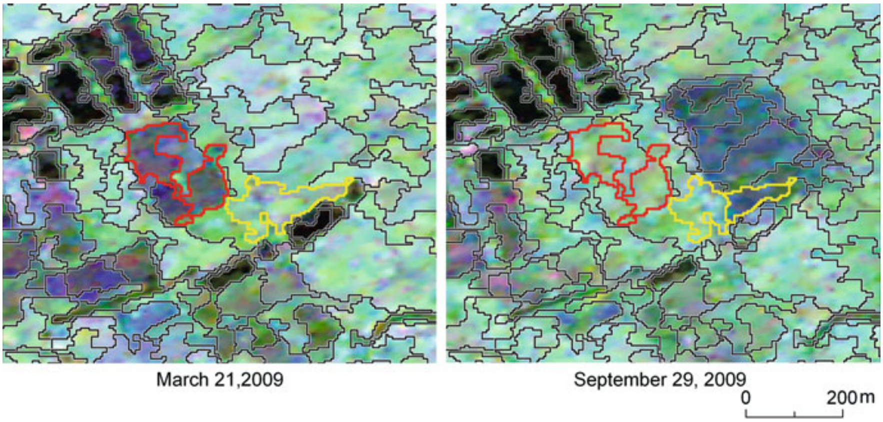


Fig. 2 Hierarchical segmentation of multi-temporal PolSAR images to delineate image objects for change detection

3.3 Object-Oriented CVA for PolSAR Images

Compared with single-polarization SAR data, PolSAR images provide different polarizations that are important for identifying different ground targets. Therefore, change detection that uses PolSAR data should utilize different polarizations. CVA is a widely used technique for change detection in the remote sensing field (Malila 1980). CVA can process any number of image channels and can produce detailed change detection information based on the channel change vector obtained by subtracting corresponding image channels of two images acquired at different times. However, the application of CVA in PolSAR images remains an open issue, and CVA is commonly used on multispectral images to detect changes at the pixel level, which is not appropriate for PolSAR images because of the speckle effect. In this study, CVA was applied on RADARSAT-2 PolSAR images to detect LULC changes at the object level. In object-oriented CVA, change detection is based on feature change vectors (FCVs), which are obtained by subtracting corresponding feature vectors of image objects in two images acquired at different dates. Two co-registered images, image (t_1) and image (t_2), are assumed to be acquired over the same area at different times t_1 and t_2. If k features are extracted from an image object, the feature vectors of the image object in the two images are given by $X = (x_1, x_2, \ldots, x_k)^T$ and $Y = (y_1, y_2, \ldots, y_k)^T$ respectively, the FCVs is defined as

$$\Delta G = X - Y = \begin{pmatrix} x_1 - y_1 \\ x_2 - y_2 \\ \cdots \\ x_k - y_k \end{pmatrix} \tag{1}$$

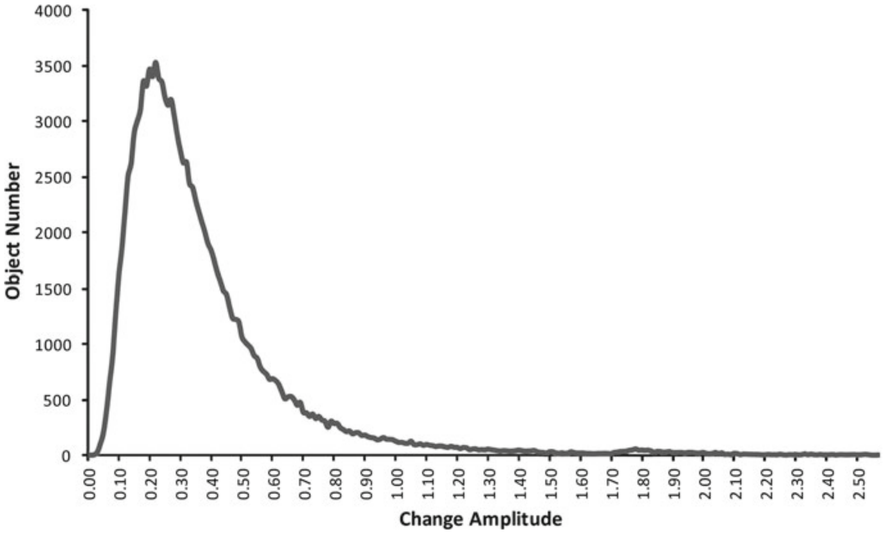


Fig. 3 Statistical distribution of the image objects in the change magnitude calculated using object-oriented CVA

where ΔG includes all the change information between the two images for a given image object, and the change magnitude $||\Delta G||$ is computed with

$$\|\Delta G\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_k - y_k)^2} \tag{2}$$

The higher the $||\Delta G||$ is, the more likely that changes take place. Unsupervised classifiers or threshold methods are commonly applied on the change magnitude to identify changes. Two aspects are essential for object-oriented CVA: one is the selection of appropriate features to calculate FCVs, and the other is the determination of a suitable threshold to identify changed objects. CVA has been applied to the backscattering matrix of PolSAR to detect the extent of change caused by an inundation (Shen et al. 2007). Given that the coherency matrix provides more information than the backscattering matrix, object-oriented CVA was implemented using the mean values of the elements of the coherency matrix of RADARSAT-2 PolSAR data to detect LULC changes. A widely accepted assumption is that the statistical distribution of the pixels of change and no-change areas in the change magnitude can be approximated as a mixture of Gaussian distributions (Bovolo and Bruzzone 2007; Camps-Valls et al. 2008). The statistical distribution of the image objects in the change magnitude, calculated using object-oriented CVA, is shown in Fig. 3. The distribution of the image objects in the change magnitude can also be approximated as a mixture of Gaussian distributions. Therefore, the expectation-maximization (EM) algorithm was applied on the change magnitude to identify changed objects. EM is frequently used for data clustering in machine learning and computer vision because it finds clusters by determining a mixture of Gaussians that fit a given data set.

3.4 Change Type Determination Using Post-Classification Comparison

The type of changed objects can be determined by using post-classification comparison of PolSAR images. Many classification methods for PolSAR data have been explored by researchers (Rignot et al. 1992; Chen et al. 1996; Cloude and Pottier 1997; Lee et al. 1999; Barnes 1988, 2007; Shimoni et al. 2009). However, so far most of the classification methods are pixel-based. These methods are prone to be affected by speckles in PolSAR images and are hard to utilize textural and spatial information of PolSAR images. Moreover, they cannot take fully use of polarimetric information of PolSAR data for LULC classification. In this work, a new method was proposed for the classification of PolSAR images based on the classification method (Qi et al. 2012), which integrates polarimetric decomposition, PolSAR interferometry, object-oriented image analysis, and decision tree algorithms. In the proposed method, PolSAR interferometry was not used and the combination of decision tree algorithms and SVMs was used to implement classification. First, different polarimetric techniques were used to extract polarimetric parameters to support classification. Second, object-oriented image analysis was used to extract textural and spatial features from image objects to improve classification accuracy. Meanwhile, the use of object-oriented image analysis could reduce the effect of speckles. Third, a decision tree algorithm was applied to select features for classification. Finally, the final LULC classification was conducted using SVMs based on the selected features. A comparison between the classification using decision tree algorithms and that based on the combination of decision tree algorithms and SVM was performed (Fig. 4). The accuracy evaluation of the classification results are shown in Tables 3 and 4. The comparison shows that combining decision tree algorithms and SVMs can achieve higher classification accuracy than decision tree algorithms.

4 Results and Discussion

Comparison between the proposed method and the PCC which is based on the supervised Wishart classifier (Lee et al. 1994) was made to test the performance of the proposed method. The supervised Wishart classifier has been commonly used for the classification of PolSAR data. This method is a pixel-based maximum likelihood classifier based on the complex Wishart distribution for the polarimetric coherency matrix. Detection accuracy, false alarm rate, and overall error rate are commonly used statistics for evaluating change detection results. The detection accuracy is the

Table 3 Classification accuracy of the combination of decision tree algorithms and SVMs

Classified data	Reference data								
	B	BS	F	L	CN	UB	W	Total	UA (%)
B	36,732	0	1,089	0	4,163	545	0	42,529	86.37
BS	107	13,255	0	5,088	457	0	5,207	24,114	54.97
F	6,564	0	20,315	25	9,174	359	0	36,437	55.75
L	0	3,353	0	29,640	0	15	1,151	34,159	86.77
CN	5,642	146	20,562	0	56,314	2,006	0	84,670	66.51
UB	738	0	2,465	0	1,618	61,118	0	65,939	92.69
W	0	312	0	2,456	0	14	63,348	66,130	95.79
Total	49,783	17,066	44,431	37,209	71,726	64,057	69,706	353,978	
PA (%)	73.78	77.67	45.72	79.66	78.51	95.41	90.88		
OA (%)	79.30								
Kappa	0.75								

Table 4 Classification accuracy of decision tree algorithms

Classified data	Reference data								
	B	BS	F	L	CN	UB	W	Total	UA (%)
B	35,523	0	2,495	0	4,059	452	0	42,529	83.53
BS	107	11,385	0	6,686	457	0	5,479	24,114	47.21
F	5,076	389	17,328	25	11,236	2,383	0	36,437	47.56
L	0	921	236	32,458	0	15	529	34,159	95.02
CN	6,273	1,335	19,041	313	53,297	4,411	0	84,670	62.95
UB	738	0	3,016	52	1,413	60,720	0	65,939	92.09
W	0	212	0	4,774	0	0	61,144	66,130	92.46
Total	47,717	14,242	42,116	44,308	70,462	67,981	67,152	353,978	
PA (%)	74.45	79.94	41.14	73.26	75.64	89.32	91.05		
OA (%)	76.80								
Kappa	0.72								

percentage of correctly labeled “change” samples. The false-alarm rate is the percentage of erroneously labeled “no-change” samples. The overall error rate is the percentage of erroneously labeled validation samples.

The change magnitude of the coherency matrix was calculated using object-oriented CVA (Fig. 5c). The EM algorithm was used to identify changed objects from the change magnitude. The change detection results of the proposed method and the PCC which is based on the Wishart supervised classification are shown in Fig. 6, and the accuracy evaluations are shown in Table 5. The overall error rate of the proposed method was 5.09 %, much lower than that of the PCC based on the Wishart supervised classification, which exhibited an overall error rate of 30.24 %. Although the detection accuracy of the proposed method was lower than that of the PCC based on the Wishart supervised classification, the proposed method significantly improved the false alarm rate of the PCC based on the Wishart supervised classification by 35.59 %.

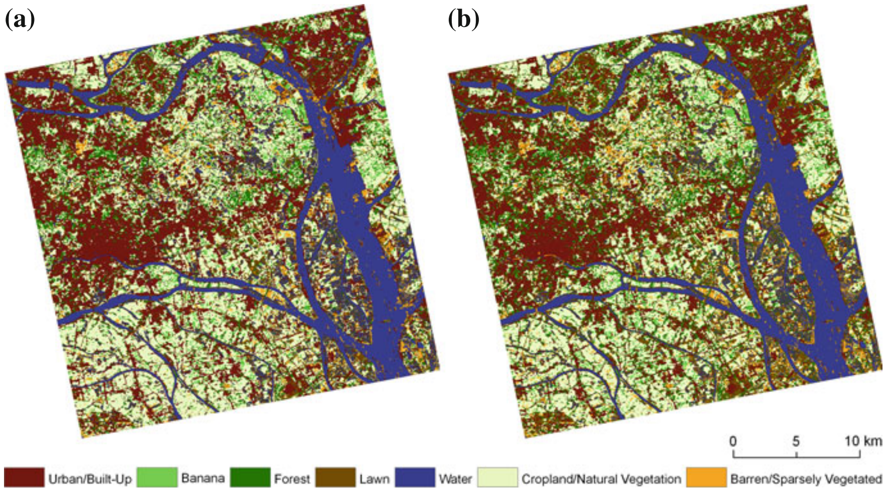


Fig. 4 Classification results: **a** combination of SVMs and decision tree algorithms; **b** decision tree algorithms

Table 5 Comparison of change detection accuracy between the proposed method and the PCC that is based on the Wishart supervised classification

	Proposed method	PCC based on the Wishart supervised classification
Detection accuracy (%)	88.85	94.87
False alarm rate (%)	3.06	38.65
Overall error rate (%)	5.09	30.24

The type of changes was determined by comparing the classification results of the two images. The result on change type is shown in Fig. 7, and the accuracy evaluations are given in Tables 6 and 7. The comparison between the proposed method and the PCC that is based on the Wishart supervised classification shows that the former achieved higher overall accuracy and kappa value in determining the type of changes. Furthermore, the user's and the producer's accuracies for most of the change types were improved when the proposed method was used. The proposed method achieved much higher user's and producer's accuracies for the change from barren/sparsely vegetated land to urban/built-up areas, from croplands/natural vegetation to urban/built-up areas, and from croplands/natural vegetation to barren/sparsely vegetated land. The detection of these LULC change types are critical in the monitoring of illegal land development. The results show that a significant improvement was achieved using the proposed detection method compared with the Wishart supervised classification-based PCC.

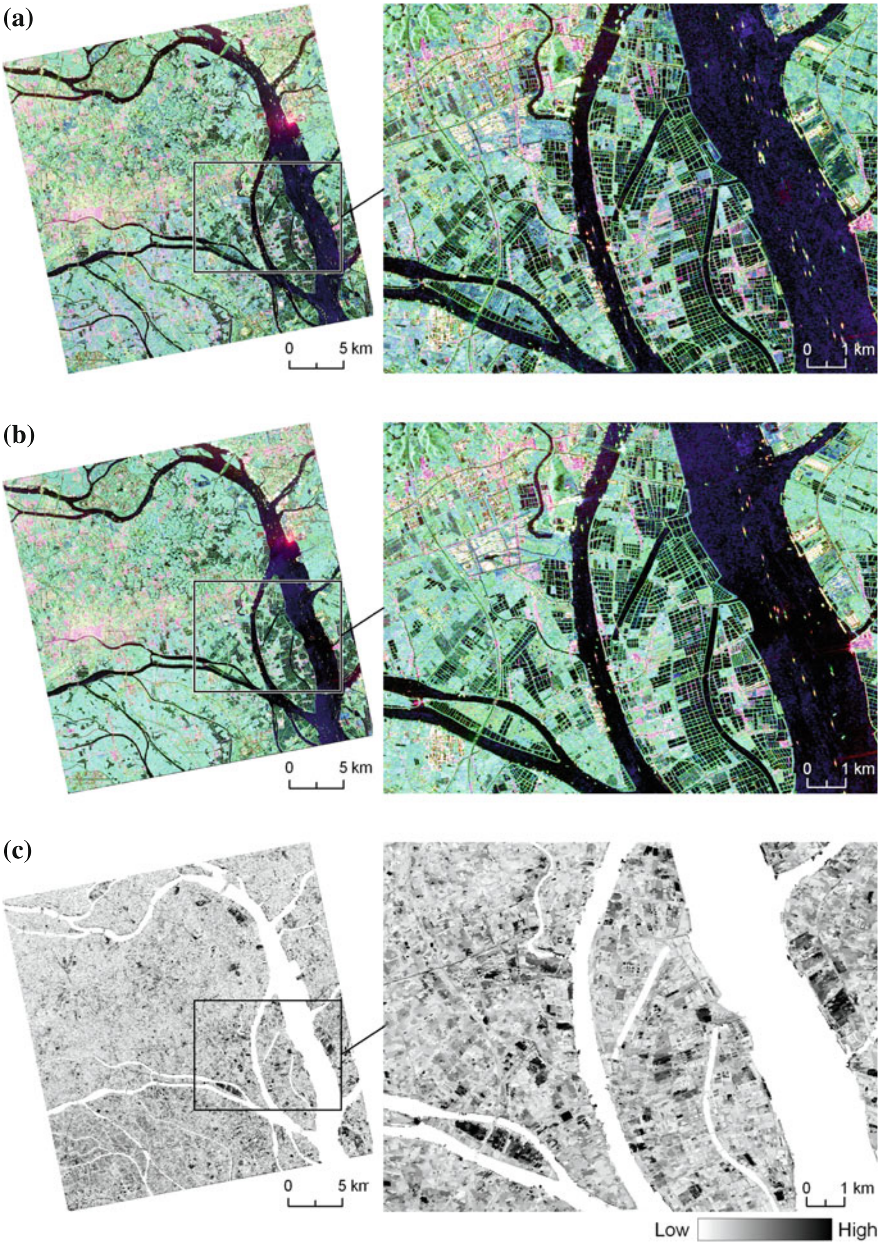


Fig. 5 **a** RADARSAT-2 image acquired on March 21, 2009; **b** RADARSAT-2 image acquired on September 29, 2009; **c** change magnitude calculated using CVA

Table 6 Accuracy of change type determination using the proposed method

Classified	Reference data																								
data	NC	B-W	BS-B	BS-F	BS-CN	BS-UB	BS-W	F-UB	L-F	L-CN	L-UB	CN-B	CN-BS	CN-UB	CN-W	UB-BS	UB-L	UB-CN	UB-W	W-BS	W-L	W-CN	W-UB	Other	Total	UA (%)
B-W	33	1,708	0	0	0	0	0	0	0	0	0	0	90	0	56	0	0	0	0	0	0	0	0	393	2,280	74.91
BS-CN	3,524	0	254	6,248	9,515	3,040	0	0	3,368	2,585	299	16	0	497	0	0	0	0	0	7	1,000	5	213	1,558	32,129	29.61
BS-UB	1,500	0	0	0	11	7,165	0	50	13	4	3,185	0	0	1,616	0	0	0	0	0	42	1	0	693	61	14,341	49.96
BS-W	3,569	0	0	0	0	0	4,792	0	0	0	0	0	235	8	790	0	0	0	0	0	0	0	0	1,557	10,951	43.76
L-CN	608	0	0	0	150	0	0	0	3,127	2,515	276	0	0	0	0	0	0	5	0	0	0	212	0	455	7,348	34.23
CN-B	1,276	0	1,380	0	0	0	0	0	0	15	6	1,195	0	0	0	0	0	0	0	0	0	0	0	33	3,905	30.60
CN-BS	808	11	0	0	0	0	20	0	0	0	0	0	2,639	0	0	0	0	0	0	0	0	0	0	647	4,125	63.98
CN-UB	49	0	0	0	0	16	0	106	0	0	10	0	0	357	0	0	0	0	0	0	0	0	0	0	538	66.36
CN-W	1,244	1,106	0	0	0	0	529	0	0	0	0	0	1,275	37	10,466	0	0	0	107	0	0	0	0	3,170	17,934	58.36
UB-BS	175	0	0	0	0	0	0	0	0	0	0	0	0	0	0	651	526	0	365	0	0	0	0	0	1,717	37.91
UB-CN	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	40	95.00
W-BS	154	0	0	0	13	0	0	0	0	66	29	0	0	0	0	8	0	0	0	2,595	863	74	9	123	3,934	65.96
W-L	319	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3,737	0	0	33	4,089	91.39
W-CN	432	0	0	0	242	0	0	0	623	992	1,064	0	0	0	0	0	0	25	0	408	43	7,353	5,360	2,912	19,454	37.80
Total	13,693	2,825	1,634	6,248	9,931	10,221	5,341	156	7,131	6,177	4,869	1,211	4,239	2,515	11,312	659	526	68	472	3,052	5,644	7,644	6,275	10,942	122,785	
PA (%)	NA	60.46	NA	NA	95.81	70.10	89.72	NA	NA	40.72	NA	98.68	62.26	14.19	92.52	98.79	NA	55.88	NA	85.03	66.21	96.19	NA	NA		
OA (%)	44.57																									
Kappa	0.41																									

Table 7 Accuracy of change type determination using PCC based on the Wishart supervised classification

Classified	Reference data																									
data	NC	B-BS	B-L	B-W	BS-B	BS-CN	BS-UB	BS-W	F-UB	L-F	L-CN	L-UB	CN-B	CN-BS	CN-UB	CN-W	UB-BS	UB-L	UB-CN	W-BS	W-F	W-L	W-CN	Other	Total	UA (%)
B-W	0	267	405	1,204	0	0	0	7	0	0	0	0	0	34	0	130	8	7	0	0	0	0	0	218	2,280	52.81
BS-CN	2,837	0	1	0	1,563	10,400	1,356	3	139	1,968	3,798	621	326	27	293	0	5	5	92	136	694	425	349	7,091	32,129	32.37
BS-UB	621	0	0	0	899	394	4,687	0	809	195	351	2,729	160	5	1,329	0	0	0	10	48	18	20	123	1,943	14,341	32.68
BS-W	1,286	12	11	7	0	0	6	4,474	2	0	0	0	0	569	10	1,577	17	27	0	3	0	0	0	2,950	10,951	40.85
L-CN	489	0	0	0	32	386	82	0	28	1,452	2,640	296	3	0	3	0	2	0	7	17	164	19	516	1,212	7,348	35.93
CN-B	116	0	0	0	1,311	64	105	0	46	0	8	2	1,377	0	96	0	2	0	3	0	0	0	0	775	3,905	35.26
CN-BS	123	580	250	35	0	0	0	42	0	0	0	0	0	1,452	0	123	100	28	0	0	0	0	0	1,392	4,125	35.20
CN-UB	28	0	0	0	0	0	38	0	242	0	0	12	9	0	183	0	0	0	0	0	0	0	0	26	538	34.01
CN-W	150	584	349	2,278	0	0	0	1,042	0	0	0	0	0	1,418	6	7,200	78	42	0	0	0	0	0	4,787	17,934	40.15
UB-BS	38	68	47	9	0	0	0	0	0	0	0	0	0	34	0	5	562	501	17	0	0	0	0	436	1,717	32.73
UB-CN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	35	0	0	0	0	1	40	87.50
W-BS	88	0	0	0	1	21	0	1	1	36	116	16	0	0	0	0	0	0	2	1,891	191	697	605	268	3,934	48.07
W-L	379	0	0	0	0	1	2	0	0	92	28	4	0	0	0	0	0	0	0	341	231	2,909	2	100	4,089	71.14
W-CN	155	0	0	0	49	262	81	0	10	502	1,592	385	1	0	0	0	0	0	7	777	2,636	475	7,624	4,898	19,454	39.19
Total	6,310	1,511	1,063	3,533	3,855	11,528	6,357	5,569	1,277	4,245	8,533	4,065	1,876	3,539	1,920	9,035	778	610	173	3,213	3,934	4,545	9,219	26,097	122,785	
PA (%)	NA	NA	NA	34.08	NA	90.22	73.73	80.34	NA	NA	30.94	NA	73.40	41.03	9.53	79.69	72.24	NA	20.23	58.85	NA	64.00	82.70	NA		
OA (%)	37.98																									
Kappa	0.34																									

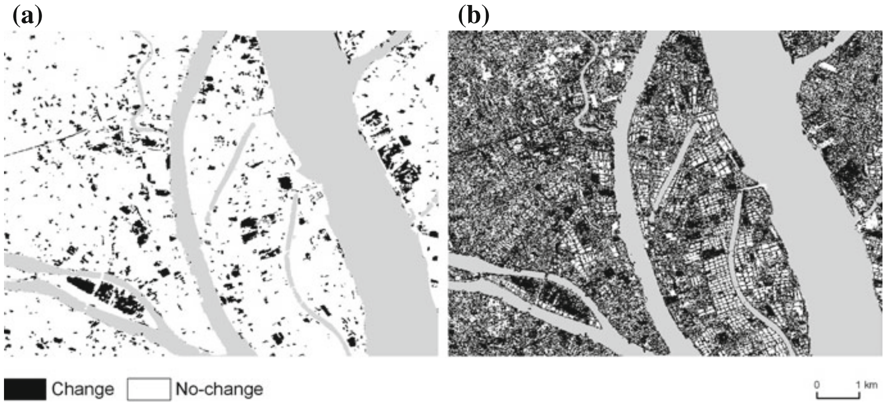


Fig. 6 Change detection result **a** proposed method; **b** PCC based on the Wishart supervised classification

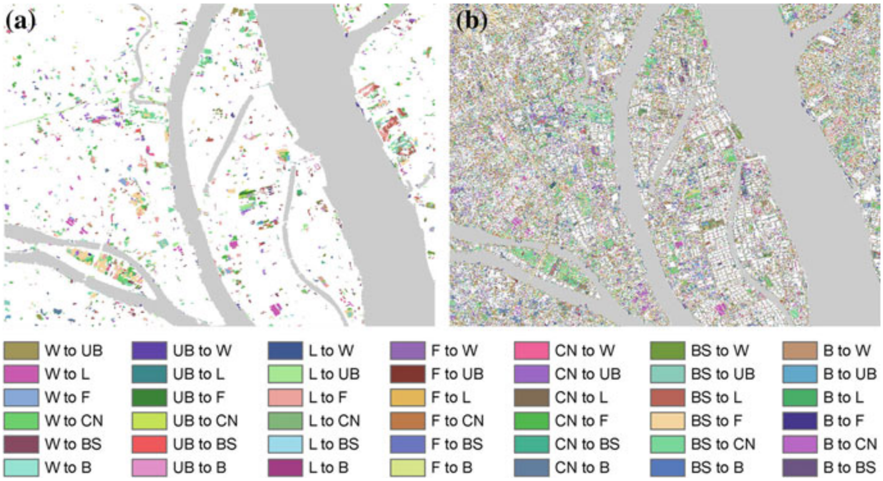


Fig. 7 Change type determination **a** proposed method; **b** PCC based on the Wishart supervised classification

5 Conclusions

This study proposed a new method for LULC change detection using RADARSAT-2 PolSAR images. The proposed method combined CVA and PCC to detect LULC changes using RADARSAT-2 images based on object-oriented image analysis. The hierarchical segmentation technique was used to delineate image objects from multi-temporal RADARSAT-2 PolSAR images for change detection. CVA was applied on the coherency matrix to identify changed objects, and then PCC was used to determine the type of changes. The classification of RADARSAT-2 PolSAR images was implemented by integrating polarimetric decomposition, object-oriented image analysis, decision tree algorithms, and SVMs. The combination of decision tree algorithms and SVMs achieved higher accuracy in the classification of PolSAR data. In comparing the proposed method and the PCC, which is based on the Wishart supervised classification, the proposed method significantly reduced overall error and false alarm rates by 25.15 and 35.59 % respectively. Moreover, the proposed method improved overall accuracy and kappa value by 6.59 % and 0.07 respectively in determining change types. The results show that the proposed method can significantly improve accuracy for LULC change detection compared with the PCC that is based on the Wishart supervised classification. Further investigations will be conducted on the selection of features used in CVA and threshold methods for identifying changes from the change magnitude calculated using CVA.

Acknowledgments This work was supported by the Science and Operational Applications Research for RADARSAT-2 Program (SOAR 2762). The authors would like to thank the Canadian Space Agency (CSA) and the MDA GEOSPATIAL SERVICES INC. for providing the RADARSAT-2 data.

References

Alberga V (2007) A study of land cover classification using polarimetric SAR parameters. Int J Remote Sens 28:3851–3870

Barnes RM (1988) Roll-invariant decompositions for the polarization covariance matrix. Polarimetry technology workshop, Redstone Arsenal, AL

Bovolo F, Bruzzone L (2007) A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain. IEEE Trans Geosci Remote Sens 45:218–236

Camps-Valls G, Gomez-Chova L, Munoz-Mari J, Rojo-Alvarez JL, Martinez-Ramon M (2008) Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection. IEEE Trans Geosci and Remote Sens 46:1822–1835

Chen KS, Huang WP, Tsay DH, Amar F (1996) Classification of multifrequency polarimetric SAR imagery using a dynamic learning neural network. IEEE Trans Geosci Remote Sens 34:814–820

Cloude SR, Pottier E (1997) An entropy based classification scheme for land applications of polarimetric SAR. IEEE Trans Geosci Remote Sens 35:68–78

Congalton R, Green K (2009) Assessing the accuracy of remotely sensed data: principles and practices. CRC Press, Boca Raton

Coppin P, Jonckheere I, Nackaerts K, Muys B, Lambin E (2004) Digital change detection methods in ecosystem monitoring: a review. Int J Remote Sens 25:1565–1596

Howarth PJ, Wickware GM (1981) Procedures for change detection using Landsat digital data. Int J Remote Sens 2:277–291

Lambin EF, Strahler AH (1994) Change-vector analysis in multitemporal space–a tool to detect and categorize land-cover change processes using high temporal-resolution satellite data. Remote Sens Environ 48:231–244

Lee JS, Grunes MR, Kwok R (1994) Classification of multi-look polarimetric SAR imagery-based on complex Wishart distribution. Int J Remote Sens 15:2299–2311

Lee JS, Grunes MR, Ainsworth TL, Du LJ, Schuler DL, Cloude SR (1999) Unsupervised classification using polarimetric decomposition and the complex Wishart classifier. IEEE Trans Geosci Remote Sens 37:2249–2258

Lee JS, Wen JH, Ainsworth TL, Chen KS, Chen AJ (2009) Improved sigma filter for speckle filtering of SAR imagery. IEEE Trans Geosci Remote Sens 47:202–213

Lu D, Mausel P, Brondizio E, Moran E (2004) Change detection techniques. Int J Remote Sens 25:2365–2407

Malila WA (1980) Change vector analysis: an approach for detecting forest changes with Landsat. In: Lafayette W (ed) Proceedings of remotely sensed data symposium

Petit CC, Lambin EF (2001) Integration of multi-source remote sensing data for land cover change detection. Int J Geogr Inf Sci 15:785–803

Qi Z, Yeh AG-O, Li X, Lin Z (2012) A novel algorithm for land use and land cover classification using RADARSAT-2 polarimetric SAR data. Remote Sens Environ 118:21–39

Rignot E, Chellappa R, Dubois P (1992) Unsupervised segmentation of polarimetric SAR data using the covariance-matrix. IEEE Trans Geosci Remote Sens 30:697–705

Shen G, Guo H, Liao J (2007) Change vector analysis method for inundation change detection using multi-temporal multi-polarized SAR images. In: Proceedings of SPIE, Wuhan, China

Shimoni M, Borghys D, Heremans R, Perneel C, Acheroy M (2009) Fusion of PolSAR and PolInSAR data for land cover classification. Int J Appl Earth Obs Geoinf 11:169–180

Silapaswan CS, Verbyla DL, McGuire AD (2001) Land cover change on the Seward Peninsula: the use of remote sensing to evaluate the potential influences of climate warming on historical vegetation dynamics. Can J Remote Sens 27:542–554

Singh A (1989) Digital change detection techniques using remotely-sensed data. Int J Remote Sens 10:989–1003

Stow DA, Tinney LR, Estes JE (1980) Deriving land use/land cover change statistics from Landsat: a study of prime agricultural land. In: Proceedings of the 14th international symposium on remote sensing of environment, Ann Arbor, Michigan

Weismiller RA, Kristof SJ, Scholz DK, Anuta PE, Momin SA (1977) Change detection in coastal zone environments. Photogramm Eng Remote Sens 43:1533–1539

Modifying Areal Interpolation Techniques for Analysis of Data on Food Assistance Benefits

Jerry Shannon and Francis Harvey

Abstract Analyses aimed at identifying food deserts—defined as areas with limited access to healthy food—have garnered much recent attention from the news media, policy makers, and non-profit groups. Much of this research relies on the proximity of large grocery stores as a measure of food access. These studies have been limited by poor data quality, boundary effects, and scale dependence. Drawing on data from the Supplemental Nutrition Assistance Program (SNAP, formerly known as food stamps), we suggest an alternative approach that incorporates the distribution and redemption of food assistance benefits in low-income neighborhoods. This data is publically available, but at the zip code level, limiting its usefulness for neighborhood analysis. We use a three-class areal interpolation method to develop three disaggregation techniques that increase the usability of this data. These utilize several external data sources to weight the distribution of this data, including the U.S. Census, U.S. Geological Survey satellite imagery, and existing cadastral data. Our analysis, focused on the Twin Cities metropolitan region for federal fiscal year 2010, thus allows for a more accurate depiction of how residents actually access the food system.

1 Introduction

Spatial analysis focused on the identification of "food deserts" has been the subject of increased research along with media and political attention in recent years, despite the fact that the concept is itself only a little more than a decade old (Clarke et al. 2004; Wrigley 2002). In 2008, the U.S. Congress passed legislation funding study of food deserts, defining them as "an area in the United States with limited access to

J. Shannon (✉) · F. Harvey
University of Minnesota, Minneapolis, MN, USA
e-mail: shann039@umn.edu

F. Harvey
e-mail: fharvey@umn.edu

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_9,

affordable and nutritious food, particularly such an area composed of predominantly lower income neighborhoods and communities" (USDA 2009, p. 1). Although the term "desert" suggests that these neighborhoods lack food of any kind, this definition does not frame the problem as a general absence of any food sources, or even the absence of healthy foods such as fruits and vegetables. Rather, as the above definition indicates, food deserts are low-income neighborhoods that lack affordable or accessible options for healthy food but a comparative abundance of highly processed, nutrient poor foods. Drawing from a broader framework of social ecological studies in public health, such neighborhoods are assumed to foster poor habits of food consumption among their residents and consequentially higher incidences of diet-related health problems such as heart disease or diabetes (Egger and Swinburn 1997; Stokols 1995; Swinburn et al. 1999).

The exact metrics used to measure food deserts have varied. A recent review of research in this area classified them using two broad categories: geographic studies and market-basket studies (Beaulac et al. 2009). Geographic studies have mainly measured food access through GIS-based analyses of distances to healthy and unhealthy food sources, frequently grocery stores and fast food restaurants respectively (Black et al. 2011; Hemphill et al. 2008; Zenk et al. 2005). While most geographic studies record the distribution of food sources throughout a given region, other researchers have prepared market-basket studies that focus on the quality of food offerings within each store (Block and Kouba 2007; Goldsberry et al. 2010; Hendrickson et al. 2006). Analyzing food prices and quality across neighborhoods helps with the identification of disparities in access to healthy foods. Despite their popularity, poor data quality, boundary effects, and scale dependence can limit these studies (Powell et al. 2011; Short et al. 2007). Since stores are often located along commercial corridors used as the boundaries of administrative units, boundary effects can be particularly problematic.

The research in this article attempts to address this last issue with current food desert research in two ways. First, by using data on actual food procurement by urban residents, we hope to avoid an over reliance on absolute measures of distance as proxies for food access. For this project, we use data from the Supplemental Nutrition Assistance Program (SNAP), also known as food stamps, the primary federal food assistance program in the United States. Data on both the distribution of SNAP benefits and their redemption at authorized food vendors is publically available, and it provides another method by which to analyze the types of stores utilized by neighborhood residents. Identification of neighborhoods with high net inward or outward flows of SNAP benefit dollars also provides another way to identify areas with low food accessibility. Second, we use areal interpolation to disaggregate this data and perform a fine scale neighborhood analysis. For this, we drew from techniques to develop three approaches to disaggregating this data. The finer scale data resolution that results from this approach significantly lessens the impact of boundary effects on data analysis and allows for a more robust, multi-scalar analysis of this data. While the results of this research are most immediately applicable to the U.S. context, this approach may serve as a broader model for the analyses of low food access.

2 Study Background

2.1 Background on the Supplemental Nutrition Assistance Program

Data on the distribution and redemption of federal food assistance provides insight on how food is procured in low-income neighborhoods. In 2010, SNAP provided assistance to over 36 million people throughout the United States (USDA 2011). In Minnesota, specifically the seven county Twin Cities Metropolitan Area, where this research is conducted, SNAP provides $29 million of benefits to approximately 270,000 individuals, and is accepted in just over 1,400 retail locations. Data on benefit distribution and redemption is publicly available through the United States Department of Agriculture (USDA) and individual state departments of public health.

2.2 Obstacles to Analysis of SNAP Data

For this project, we obtained data on monthly SNAP benefit distribution and redemption for October 2009 through September 2010. However, two main obstacles to analysis quickly became apparent. First, while we were able to acquire data on benefit distribution and benefit redemption, these two datasets were held in different locations. Client and distribution data are managed by the Minnesota Department of Health, while vendor and benefit redemption data are held at the federal level by the U.S. Department of Agriculture. These two data sets follow different data models with the consequence that there is no clear way to link benefit distributions to clients' eventual redemptions at stores. Following data protection guidelines and to protect stores' exact sales numbers, the USDA releases data on benefit redemptions only for zip codes in which four or more retailers are present. Second, zip codes are the finest scale at which this data are offered (Fig. 1). While this scale may be sufficient for a broad analysis, it makes meaningful analysis at the neighborhood level difficult (Raper et al. 1992).

2.3 Areal Interpolation and Dasymetric Mapping

Areal interpolation offers a useful spatial analytical tool set to develop more detailed analysis of food deserts, as it allows for very small area estimation of food benefit distribution and redemption. Broadly speaking, areal interpolation refers to the reclassification of data from one set of areal units to another (Flowerdew and Green 1991; Goodchild and Anselin 1993; Goodchild and Lam 1980). Much recent work in this area has focused on the use of dasymetric mapping techniques While it has been in use for over a century, dasymetric maps have recently enjoyed greater usage among

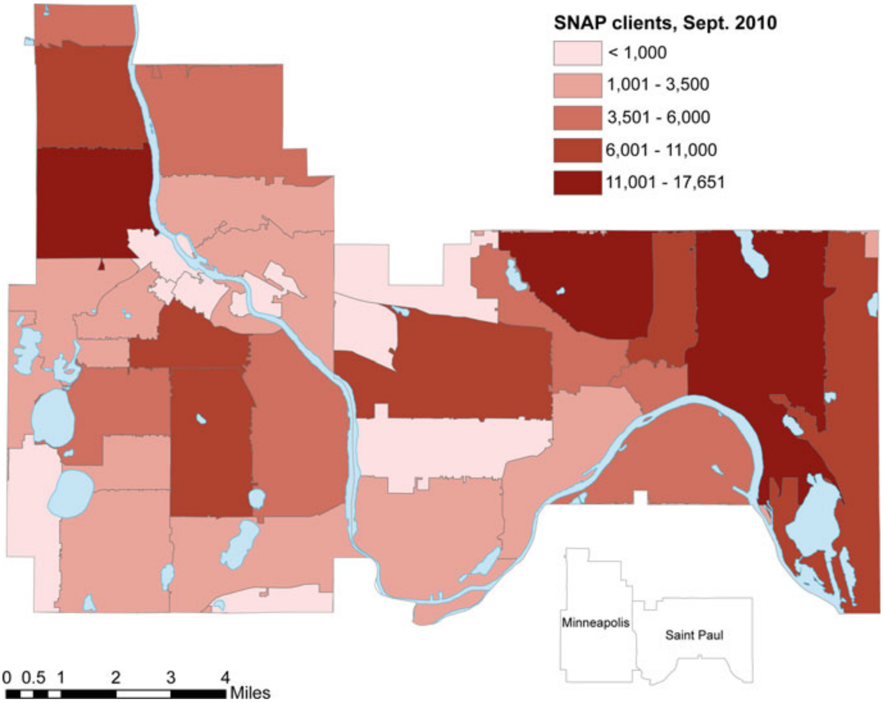


Fig. 1 Number of SNAP recipients per zip code in Minneapolis and St. Paul, September 2010

population geographers. In contrast to choropleth maps, which display variables as an uniform distribution within often politically determined areal units, an analysis using dasymetric maps highlights continuities of a given population variable over space. Thus, boundaries on dasymetric maps represent meaningful changes in the variable of interest, unlike choropleth maps, where boundaries are generally unrelated to these variables (Fotheringham and Rogerson 1993; O'Sullivan and Unwin 2002; Wright 1936). The distinction between areal interpolation and dasymetric mapping has, in practice, been somewhat fuzzy (Mennis 2009), and by seeking to develop small area estimations of food benefit utilization the approach outlined here borrows from both techniques.

Papers by Eicher and Brewer (2001) and Mennis (2003) recently have described several techniques by which datasets aggregated to political units, such as census data, might be transformed to a small area map. Both these papers advocate the use of land use classifications drawn from remotely sensed imagery to weigh the distribution of populations within pre-defined areal units, though the exact nature of the distributional method varies. Eicher and Brewer found that a limiting variable method, where land types are capped at a certain population density, produced the most accurate results. Eicher and Brewer found greater errors in the three-class method, which weights distribution of a demographic variable based on an underlying characteristic from an auxillary dataset such as land-use type. Mennis developed a modified version of the three-class method that addresses their concerns. Using classified land imagery and block-group level census data, this technique averages population densities in various land use types. These densities are then combined with area measurement to weight the distribution of that uniformly distributed block group data to a finer-scale raster distribution.

Applied research in a range of areas has drawn upon this approach (Langford 2006; Poulsen and Kennedy 2004). Satellite imagery is the most commonly used external dataset used for weighting. In the United States, preclassified land imagery from the National Land Cover Dataset (NLCD) from the U.S. Geological Survey simplifies this approach (Reibel and Agrawal 2007). Maantay et al. (2008) develop their own dasymetric mapping system based on cadastral data including residential area and number of units per residence. Other recent researchers have suggested alternative population estimation methods using automated classification or street network density (Langford 2006; Reibel and Bufalino 2005; Tapp 2010). However, each of these techniques is best suited for estimates of general population. As this study is focused particularly on the population of individuals receiving food assistance, a three-class method relying on a weighting variable from survey data was deemed most appropriate.

3 Methods

Our method adapted areal interpolation techniques to three different scenarios (outlined in Table 1). These involved a variety of areal units: polygons, points, and raster cells. To weight distribution of the data, we used both external datasets and averages of existing data. For data on benefit distribution, three external datasets were used to weight the disaggregation: (1) zoning classifications for the Twin Cities Metropolitan Area provided by a regional governmental council, (2) NLCD preclassified land use imagery, and (3) demographic data from the U.S. Census' American Community Survey (ACS) for the 5 year summary period 2005–2009, the most recent available at the time of this research. For benefit redemptions, USDA data on existing SNAP redemption patterns were used. Thus, while these three steps used similar methods, each required a specific adaptation of existing methods.

3.1 Benefit Distribution: From Zip Code to "Ziptracts"

Initial attempts to disaggregate this data directly to a density raster resulted in artificially sharp breaks at zip code boundaries, and so we developed a two stage process in which data is first disaggregated to the census tract level using ACS data and then further disaggregated to a 30 x 30 m grid using zoning and land use classifications.

Table 1 Three stages of disaggregation in our project

	Benefit distribution, point 1	Benefit distribution, point 2	Benefit redemptions
Data source	Minnesota Department of Health	Minnesota Department of Health	USDA
Initial resolution	Zip codes	"Ziptracts"	Zip codes
Ending resolution	"Ziptracts"	30 × 30 m raster	Points (store locations)
Population weighting variable	Density of SNAP households	Average density in existing data for 20 land use classes	SNAP redemption patterns by store type
Source of weighting areal units	U.S. Census	USGS preclassified land use imagery, regional zoning data	USDA
Data resolution	Census tracts	30 × 30m raster	N/A (weights are not spatial)
Area ratio utilized	Yes	Yes	No

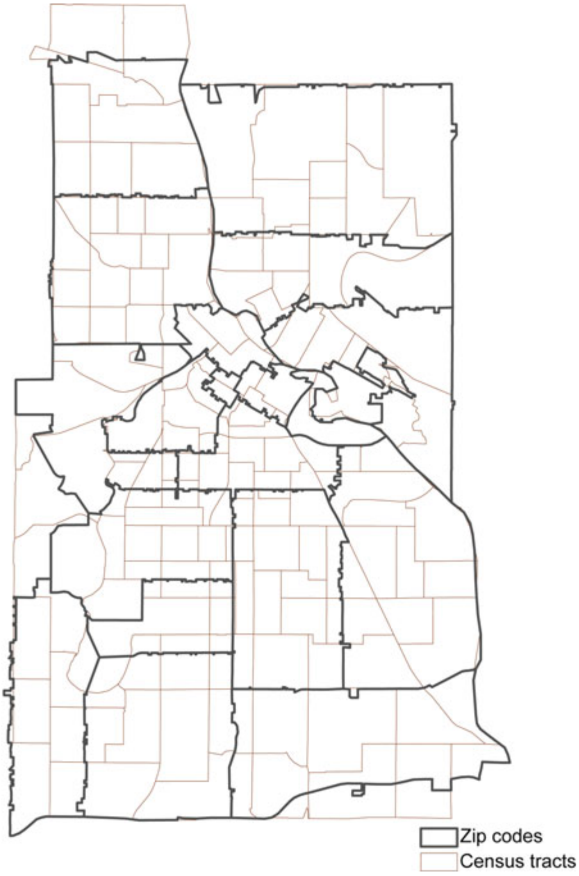


Fig. 2 Zip code and tract boundaries in Minneapolis

Zip codes and census tracts vary in size, though the former are generally much larger than the latter in high-density urban areas (Figs. 2, 3).

The steps of our analysis can be done in most GIS software packages. Prior to the first stage of this process, both the zip code and tract layers were clipped so that they only included land zoned for permanent residential use. We calculate food stamp utilization, reported by households in the ACS, as a density based on the clipped area of each census tract (per km^2). Finally in the analysis we create a set of unique polygons that share the same zip code and tract identifiers (Fig. 4). To distinguish them, the new polygons of this layer were referred to as "ziptracts."

Following the guidelines outlined in Mennis (2003), a population fraction and area ratio are created for each zip tract. The population fraction is based on the ACS household density, normalized against all tracts within a given zip code. In a hypothetical zip code A containing tracts 1, 2, and 3, the population fraction would be represented in this way:

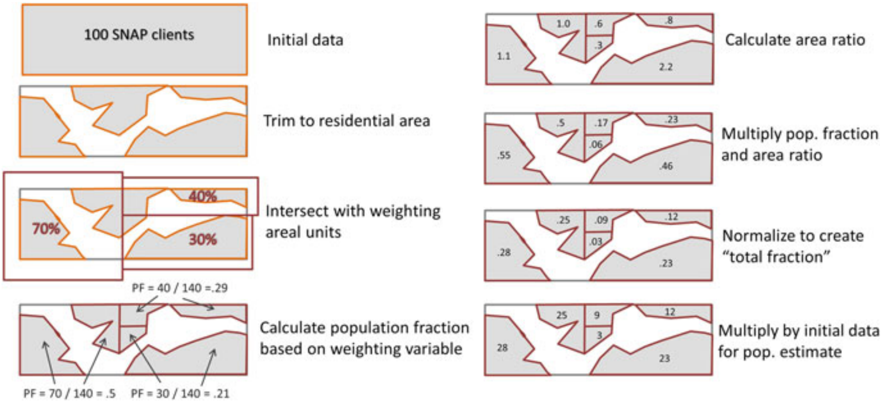

100 SNAP clients
Initial data
Trim to residential area
70%
40%
30%
Intersect with weighting areal units
PF = 40 / 140 =.29
Calculate population fraction based on weighting variable
PF = 70 / 140 = .5
PF = 30 / 140 = .21
1.0
.6
.8
1.1
.3
2.2
Calculate area ratio
.5
.17
.23
.55
.06
.46
Multiply pop. fraction and area ratio
.25
.09
.12
.28
.03
.23
Normalize to create "total fraction"
25
9
12
28
3
23
Multiply by initial data for pop. estimate

Fig. 3 Overview of approach used to disaggregate to "ziptracts"

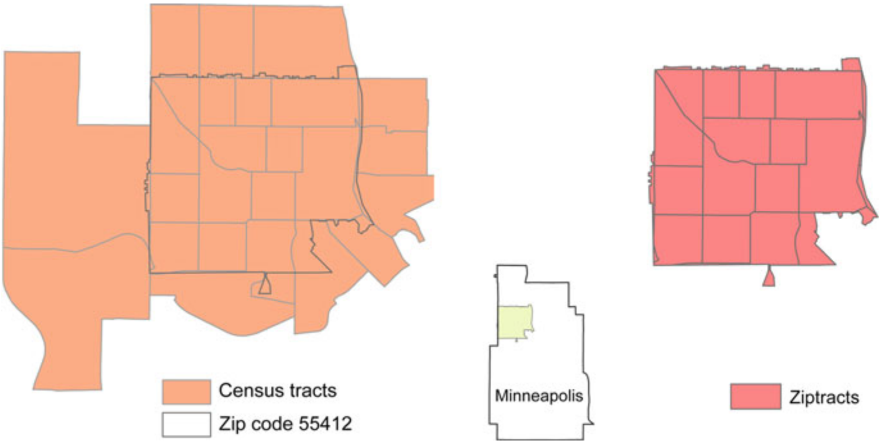


Fig. 4 Census tracts and the boundary of zip code 55412 (*left*) and results of intersecting these two boundaries (*right*)

$$\mathrm{pf}_{A1} = \frac{\mathrm{den}_1}{(\mathrm{den}_1 + \mathrm{den}_2 + \mathrm{den}_3)}$$

In this case, pf_{A1} refers to the population fraction for the ziptract for zip code A and tract 1, den_1 refers to the weighting variable (rate or density from the ACS) for tract 1, and den_2 and den_3 refer to the weighting variables for tracts 2 and 3.

The area ratio adjusts for the unequal areas of each tract within a zip code. It compares the actual proportion of a tract's area in a zip code to its expected proportion if all areas were equal. If a zip code contained three tracts, for example, that expected proportion would be 0.33 for each tract (see Mennis (2003), for further explanation). The area ratio in this instance would be greater than 1 for tracts taking up more than one-third of a zip code's total area or less than one for those smaller than a third of the zip code's area. Using the hypothetical example given above, the area ratio can be written:

$$\mathrm{ar}_{\mathrm{A1}} = \left(\frac{\mathrm{area}_{\mathrm{A1}}}{\mathrm{area}_{\mathrm{A}}}\right) \Big/ \left(\frac{1}{\#\mathrm{tracts}_{\mathrm{A}}}\right)$$

Here, $\mathrm{ar}_{\mathrm{A1}}$ refers to the area ratio for the ziptract for zip code A and tract 1, $\mathrm{area}_{\mathrm{A1}}$ is the area of that ziptract, $\mathrm{area}_{\mathrm{A}}$ is the total area of that zip code, and $\#\mathrm{tracts}_{\mathrm{A}}$ is the total number of tracts in zip code A.

Once computed, the population fraction and area ratios are multiplied together and then again normalized to determine a total fraction for each ziptract. The calculation would be written in this way:

$$\mathrm{tf}_{\mathrm{A1}} = \frac{(\mathrm{pf}_{\mathrm{A1}} \times \mathrm{ar}_{\mathrm{A1}})}{((\mathrm{pf}_{\mathrm{A1}} \times \mathrm{ar}_{\mathrm{A1}}) + (\mathrm{pf}_{\mathrm{A2}} \times \mathrm{ar}_{\mathrm{A2}}) + (\mathrm{pf}_{\mathrm{A3}} \times \mathrm{ar}_{\mathrm{A3}}))}$$

Here, $\mathrm{tf}_{\mathrm{A1}}$ refers to the total fraction for the ziptract for zip code A and tract 1, $\mathrm{pf}_{\mathrm{A1}}$ is the population fraction and area ratios for the tracts 1, 2, and 3 are listed as above. Once calculated, the recorded population of SNAP clients for the zip code in the benefit distribution data is multiplied by this total fraction to determine the estimated number of clients living within the ziptract.

3.2 *Benefit Distribution: From Ziptract to Raster Cell*

Once our benefit distribution data is disaggregated to the ziptract scale, we use zoning and land use classification data to create a density raster. The zoning data contain five residential classifications (farmstead, single family detached, single family attached, multifamily, and manufactured housing). This data is converted from vector to a $30 \times 30\,\mathrm{m}$ raster so that both datasets have the same formatting and resolution. We clip the land cover layer to match the extent of the zoning data, resulting in four classifications (open land, urban-light use, urban-medium use, and urban-heavy use). We then combine these two rasters to create a new layer containing 20 distinct classifications (farmstead-open land, farmstead-light use, etc.). These datasets are complementary: an area on the urban fringe zoned single family detached has a very different density than the same zoning category in the urban core.

Since no population variable could be used to weight data at this scale, we then calculate our own density weights based on our ziptract data. The density of SNAP participants was calculated for each ziptract by dividing our estimated count by the ziptract's area. Using the ArcGIS zonal statistics tool, we then determined the average density of each of our 20 classifications. These densities then became the weights for our disaggregation. Similar to the first stage, these weights were normalized by dividing them by the sum of weights for all classes within a zip tract. For example, in zip tract B containing land use classes 10, 20, and 30, the population fraction for land use 10 would be calculated as

$$\mathrm{pf}_{\mathrm{B}10} = \frac{\mathrm{w}_{\mathrm{B}10}}{(\mathrm{w}_{\mathrm{B}10} + \mathrm{w}_{20} + \mathrm{w}_{30})}$$

where $\mathrm{pf}_{\mathrm{B}10}$ is the population fraction for class 10, $\mathrm{w}_{\mathrm{B}10}$ is the calculated weight for class 10, and $\mathrm{w}_{\mathrm{B}20}$ and $\mathrm{w}_{\mathrm{B}30}$ are the weights for classes 20 and 30.

Zonal statistics were again used to sum the area of each classification within ziptracts and to calculate the ziptracts total area. The area ratio was calculated as in step 1 of our research:

$$\mathrm{ar}_{\mathrm{B}10} = \left(\frac{\mathrm{area}_{\mathrm{B}10}}{\mathrm{area}_{\mathrm{B}}}\right) \bigg/ \left(\frac{1}{\#\mathrm{classes}_{\mathrm{B}}}\right)$$

Where $\mathrm{ar}_{\mathrm{B}10}$ is the area ratio of class 10 in ziptract B, $\mathrm{area}_{\mathrm{B}10}$ is the area of class 10 in ziptract B, $\mathrm{area}_{\mathrm{B}}$ is the total area of ziptract B, and $\#\mathrm{classes}_{\mathrm{B}}$ is the number of classes present in ziptract B.

The "total fraction" is also calculated as in the previous stage. The estimated ziptract population was multiplied by this fraction to determine the population of each class within each zip tract. Assuming a equal dispersion, this population was then distributed within each class by dividing it by the number of cells, which is found by dividing the total class area by the cell size.

3.3 Benefit Redemption

Data on SNAP benefit redemption is also aggregated by zip code. Unlike the benefit distribution data described above, redemption data is disaggregated to discrete points. This method of moving from polygons to points was simpler, as no area weighting was needed. Following data privacy restrictions, the data set we obtained from the USDA excludes zip codes that contained 3 or less eligible locations, largely on the urban fringe or affluent neighborhoods (Names and addresses of SNAP eligible vendors are available as a downloadable spreadsheet at a USDA website (http://www.snapretailerlocator.com)). For this analysis we use a geocoded list of vendors for federal fiscal year 2010, the period of this study. We code these stores based on categories adapted from USDA's own reporting. National rates of redemption at these various store types are available, and we used these rates to weight our disaggregation (USDA 2009, p. 62). For example, in 2008, 47 % of food stamp benefits were redeemed at supermarkets, meaning that supermarkets in our scheme received a raw weighting of 0.47.

Our method of disaggregation here is similar to the population fraction described above. For each zip code, we summed the number of stores in each classification. These weights are then normalized by dividing the weight of each store by the sum of weights for all stores within a zip code. This normalized weight is then multiplied

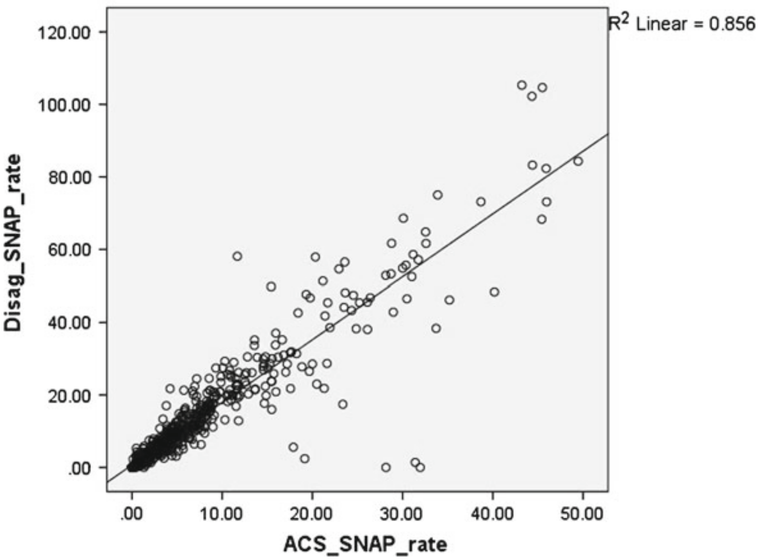


Fig. 5 Disaggregated SNAP rates for tracts measured against the reported ACS household SNAP rate

by benefit redemption amount for the zip code, with the result being an estimate redemption amount for that particular store. For store 1 in zip code A, which also contains stores 2, 3, and 4, this would be written

$$\mathrm{red}_{A1} = \mathrm{red}_{A} \times \frac{\mathrm{rw}_{A1}}{(\mathrm{rw}_{\mathrm{A1}} + \mathrm{rw}_{\mathrm{A2}} + \mathrm{rw}_{\mathrm{A3}} + \mathrm{rw}_{\mathrm{A4}})}$$

where $\mathrm{red}_{\mathrm{A1}}$ is the estimated redemptions at store 1 in zip code A, $\mathrm{red}_{\mathrm{A}}$ is the total redemption dollars in zip code A, and $\mathrm{rw}_{\mathrm{A1}}$, $\mathrm{rw}_{\mathrm{A2}}$, $\mathrm{rw}_{\mathrm{A3}}$, and $\mathrm{rw}_{\mathrm{A4}}$ are the raw weights assigned to stores 1, 2, 3, and 4 respectively.

4 Results

4.1 Disaggregation of Benefit Distribution Figures

To assess the effectiveness of this method, we reaggregate ziptracts back to the tract level, summing their estimated population of SNAP clients. Using total population from the 2010 Census, we create a ratio of SNAP clients to the general population and plotted this rate against the household participation rate in the ACS data. We expected to find a strong overall correspondence between these variables, which this analysis confirmed (Fig. 5). There were a handful of upper and lower outliers. The

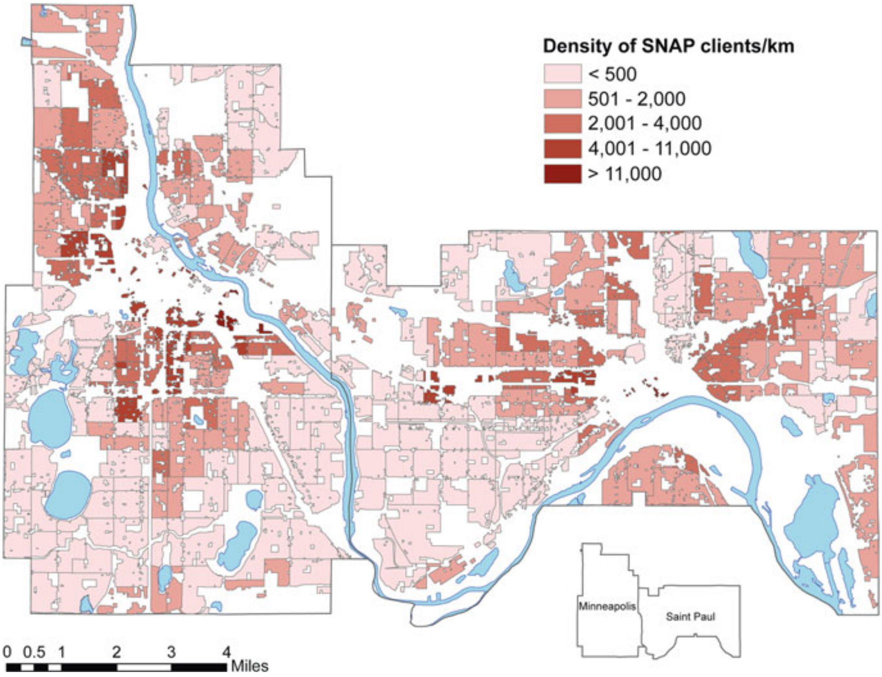


Fig. 6 Modeled density of SNAP clients per ziptract (residential areas only), September 2010

latter were explained by our data on actual SNAP participation, which in the case of lower outliers were lower than ACS estimates. The former are concentrated in three zip codes with high SNAP participation overall. They are potentially addressed through bounding the upper limit of the disaggregation, and the use of this technique will be incorporated in future research. It is also worth noting that SNAP participation numbers were significantly higher than ACS rates ($\beta = 1.73$). This factor may reflect undercounting or the effects of the economic recession, which began in 2008, as the difference between household and individual counts alone is unlikely to explain the difference.

This weighting also was more effective than just disaggregation based on area. Comparing Figs. 1 and 6, in the northwest quadrant of Minneapolis, there is much more internal heterogeneity than the zip code data alone would indicate. The east/west gradient of this data is particularly visible and there is relatively little remaining effect from the zip code boundaries in the choropleth map.

The full disaggregation using zoning and land use classifications provides further detail, with differences in estimation within census tracts clearly visible, though tract boundaries still had a significant effect (Fig. 7). In the southern section of Minneapolis, for example, the gradient from high to low values was smoothed significantly (Fig. 8).

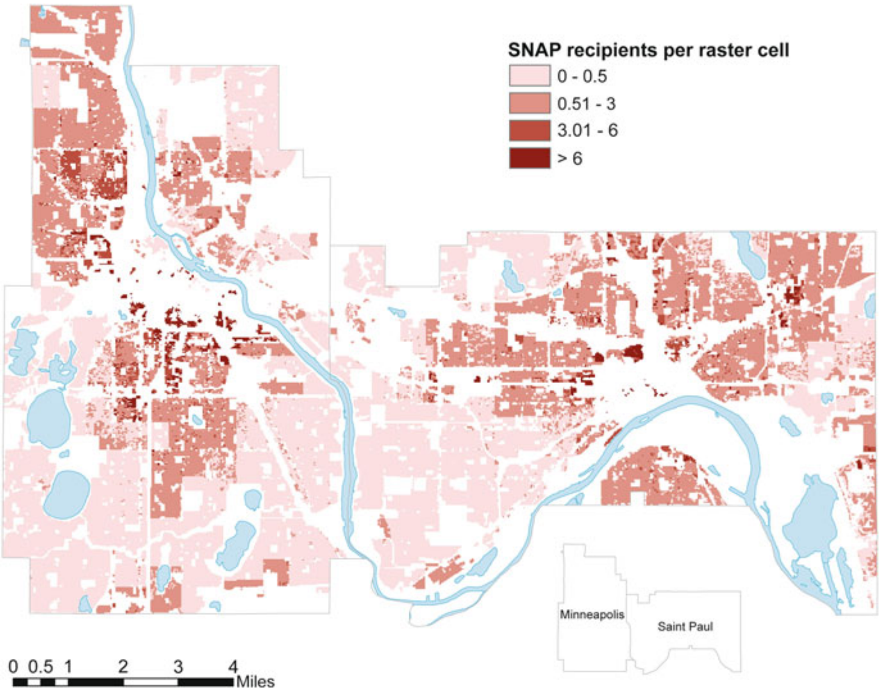


Fig. 7 Modeled density raster of SNAP clients, Sept. 2010

4.2 Disaggregation of Benefit Redemption

Disaggregating store information provides estimated benefit redemption at 1,352 stores in the Twin Cities metro area. Examining the distribution of these stores their placement along major roadways becomes apparent (Figs. 9 and 10). This highlights the potential boundary effects of analysis on zip code data and the improved usability of this dataset. By far, the largest source of redemption dollars in the Twin Cities is supermarkets (65 % of total redemptions), which is unsurprising as it was weighted most heavily in our disaggregation (Fig. 9). Convenience and corner stores represent nearly half of total stores, though they account for only 23 % of total redemptions. This is higher than national redemption rates, and further testing of this procedure might further refine this technique to better match the national sample.

5 Conclusions and Future Research

Areal interpolation provides a useful way to facilitate analysis of data on food shopping practices in low-income neighborhoods. By adapting these methods to a two-stage process for benefit distribution and using a similar procedure to estimate

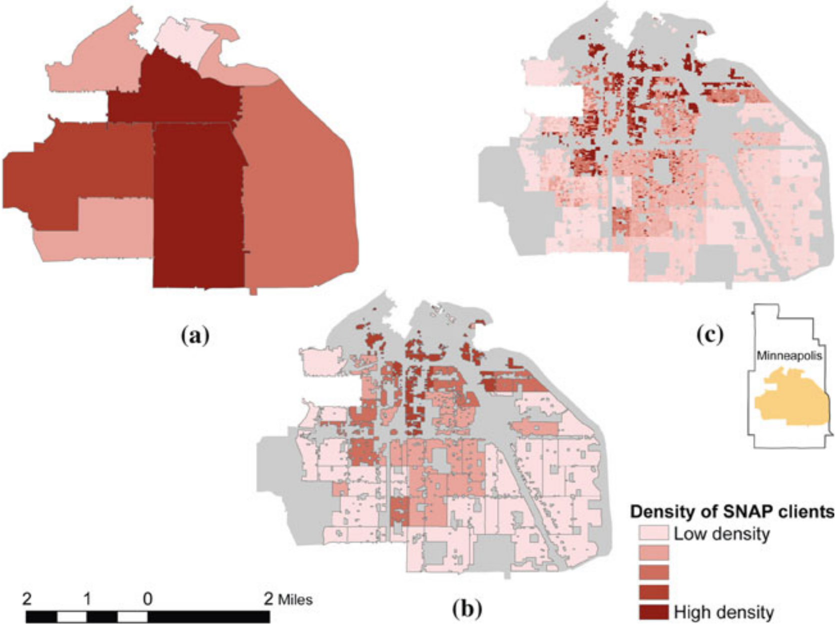


Fig. 8 Results of the two stage process of disaggregation with zip codes (**a**), ziptracts (**b**), and raster (**c**)

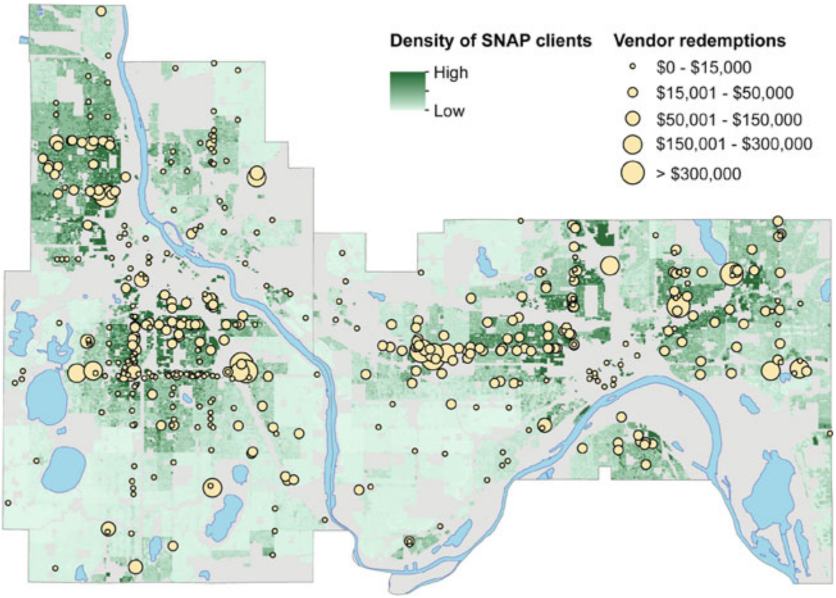


Fig. 9 Graduated symbol map of the modeled density of SNAP clients and estimated benefit redemptions in Minneapolis and St. Paul, Sept. 2010

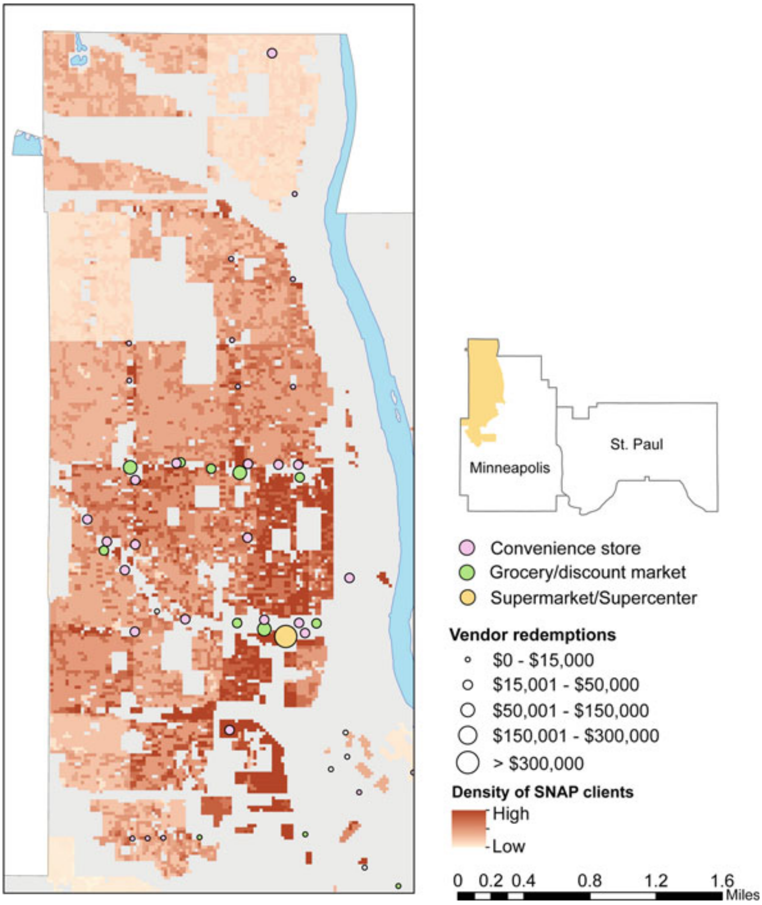


Fig. 10 Modeled density of SNAP clients and benefit redemptions by store in north Minneapolis, Sept. 2010

redemptions at individual store locations, we can begin a neighborhood-level analysis, which avoids the scalar and boundary effects of zip code data. Ground truthing the accuracy of these estimates could be a main task of future research.

Nonetheless, the initial results of our analysis have shown that the use of areal interpolation techniques in this context can contribute significantly to research on issues of neighborhood influences on food access. By producing fine scale data on the usage of food assistance programs, this method can shed light on a number of areas: the profile of food benefit distribution and usage in varying low-income neighborhoods, the relationship between food stamp usage and other measures of disparity such as poverty rate, and the effects of distance to various food outlets on household shopping patterns. More specifically, future research will aggregate benefit disbursement and redemption through checkerboard grids of decreasing size to analyze the scale at which poor access to food sources (measured as net outflows of food assistance) becomes noticeable. This data may also better demonstrate the role of small and mid-sized markets in providing access to food in low-income neighborhoods. This is particularly helpful as these stores are most common in dense urban areas and not often included in food desert measures. In sum, we find this a promising technique to advance knowledge of food deserts and their consequences.

References

Beaulac J, Kristjansson E, Cummins S (2009) A systematic review of food deserts, 1966–2007. Preventing Chronic Dis 6(3):A105

Black JL, Carpiano RM, Fleming S, Lauster N (2011) Exploring the distribution of food stores in British Columbia: associations with neighbourhood socio-demographic factors and urban form. Health Place 17(4):961–70

Block D, Kouba J (2007) A comparison of the availability and affordability of a market basket in two communities in the Chicago area. Public Health Nutr 9(7):837–845

Clarke I, Hallsworth A, Jackson P, Kervenoael RD, Perez-del-Aguila R, Kirkup M (2004) Retail competition and consumer choice: contextualising the "food deserts" debate. Int J Retail Distrib Manage 32(2):89–99

Flowerdew R, Green M (1991) Using areal interpolation methods in geographic information systems. Pap Reg Sci 70(3):303–315

Egger G, Swinburn B (1997) An "ecological" approach to the obesity pandemic. Br Med J 315:477–480

Eicher CL, Brewer C (2001) Dasymetric mapping and areal interpolation: implementation and evaluation. Cartography Geogr Inform Sci 28(2):125–138

Fotheringham A, Rogerson PA (1993) GIS and spatial analytical problems. Int J Geogr Inform Syst 7(1):3–19 (Taylor & Francis). doi:10.1080/02693799308901936

Goldsberry K, Duvall CS, Howard PH, Stevens JE (2010) Visualizing nutritional terrain: a geospatial analysis of pedestrian produce accessibility in Lansing, Michigan, USA. Geocarto Int 25(6):37–41 (Taylor & Francis)

Goodchild M, Anselin L (1993) A framework for the areal interpolation of socioeconomic data. Environ Plann A 25:383–397

Goodchild M, Lam N (1980) Areal interpolation: a variant of the traditional spatial problem. Geo-Processing 1:297–312

Hemphill E, Raine K, Spence JC, Smoyer-Tomic KE (2008) Exploring obesogenic food environments in Edmonton, Canada: the association between socioeconomic factors and fast-food outlet access. Am J Health Promotion AJHP 22(6):426–432. http://www.ncbi.nlm.nih.gov/pubmed/18677883

Hendrickson D, Smith C, Eikenberry N (2006) Fruit and vegetable access in four low-income food deserts communities in Minnesota. Agric Hum Values 23:371–383

Langford M (2006) Obtaining population estimates in non-census reporting zones: an evaluation of the 3-class dasymetric method. Comput Environ Urban Syst 30(2):161–180

Maantay JA, Maroko AR, Porter-Morgan H (2008) Research Note–a new method for mapping population and understanding the spatial dynamics of disease in urban areas: asthma in the bronx, New York. Urban Geogr 29(7):724–738

Mennis J (2003) Generating surface models of population using dasymetric mapping. Prof Geogr 55(1):31–42

Mennis J (2009) Dasymetric mapping for estimating population in small areas. Geogr Compass 3(2):727–745

O'Sullivan D, Unwin D (2002) Geographic information analysis. Wiley, Hoboken

Poulsen E, Kennedy LW (2004) Using dasymetric mapping for spatially aggregated crime data. J Quant Criminol 20(3):243–262 (Springer). doi:10.1023/B:JOQC.0000037733.74321.14

Powell LM, Han E, Zenk SN, Khan T, Quinn CM, Gibbs KP, Pugach O, et al. (2011) Field validation of secondary commercial data sources on the retail food outlet environment in the U.S. Health Place 17(5):1122–31 (Elsevier). doi:10.1016/j.healthplace.2011.05.010

Raper J, Rhind D, Shepherd J (1992) Postcodes: the new geography. Longman, Essex

Reibel M, Agrawal A (2007) Areal interpolation of population counts using pre-classified land cover data. Population Res Policy Rev 26(5–6):619–633

Reibel M, Bufalino ME (2005) Street-weighted interpolation techniques for demographic count estimation in incompatible zone systems. Environ Plann A 37(1):127–139

Short A, Guthman J, Raskin S (2007) Food deserts, oases, or mirages?: small markets and community food security in the San Francisco Bay area. J Plann Educ Res 26(3):352–364

Stokols D (1995) Translating social ecological theory into guidelines for community health promotion. Am J Health Promot 10(4):282–298

Swinburn B, Egger G, Raza F (1999) Dissecting obesogenic environments: the development and application of a framework for identifying and prioritizing environmental interventions for obesity. Prev Med 29(6):563–570

Tapp AF (2010) Areal interpolation and dasymetric mapping methods using local ancillary data sources. Cartography Geogr Inform Sci 37(3):215–228

USDA (2011) Food environment atlas. http://maps.ers.usda.gov/FoodAtlas/. Accessed 26 March 2011

USDA Economic Research Service (2009) Access to affordable and nutritious food–measuring and understanding food deserts and their consequences. Report to congress

Wright JK (1936) A method of mapping densities of population: with Cape Cod as an example. Geogr Rev 26(1):103–110

Wrigley N (2002) Food Deserts in British cities: policy context and research priorities. Urban studies (1 Oct 2002). doi:10.1080/0042098022000011344

Zenk SN, Schulz AJ, Israel B (2005) Neighborhood racial composition, neighborhood poverty, and the spatial accessibility of supermarkets in metropolitan Detroit. J Public 95(4):660–667

Representing Terrain with Mathematical Operators

Christopher Stuetzle and W. Randolph Franklin

Abstract This work describes a mathematical representation of terrain data consisting of a novel operation, the "drill". It facilitates the representation of legal terrains, capturing the richness of the physics of the terrain's generation by digging channels in the surface. Given our current reliance on digital map data, hand-held devices, and GPS navigation systems, the accuracy and compactness of terrain data representations are becoming increasingly important. Representing a terrain as a series of operations that can procedurally regenerate the terrains allows for compact representation that retains more information than height fields, TINs, and other popular representations. Our model relies on the hydrography information extracted from the terrain, and so drainage information is retained during encoding. To determine the shape of the drill along each channel in the channel network, a cross section of the channel is extracted, and a quadratic polynomial is fit to it. We extract the drill representation from a mountainous dataset, using a series of parameters (including size and area of influence of the drill, as well as the density of the hydrography data), and present the accuracy calculated using a series of metrics. We demonstrate that the drill operator provides a viable and accurate terrain representation that captures both the terrain shape and the richness of its generation.

Keywords Terrain · Operators · Representation · Compression

1 Introduction

This paper presents a novel representation of terrain data consisting of a series of mathematical operations that produce hydrographically sound *legal terrains*. These include terrains that have discontinuities on the surface (such as cliff faces and caves),

C. Stuetzle (✉) · W. R. Franklin
Rensselaer Polytechnic Institute, 110 8th St, Troy, NY12180, USA
e-mail: stuetc@cs.rpi.edu

W. R. Franklin
e-mail: mail@wrfranklin.org

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_10,

few if any local minima (pits), and whose digital formation mimics physical phenomena associated with geological terrain formation (such as erosion, digging, and faulting/folding).

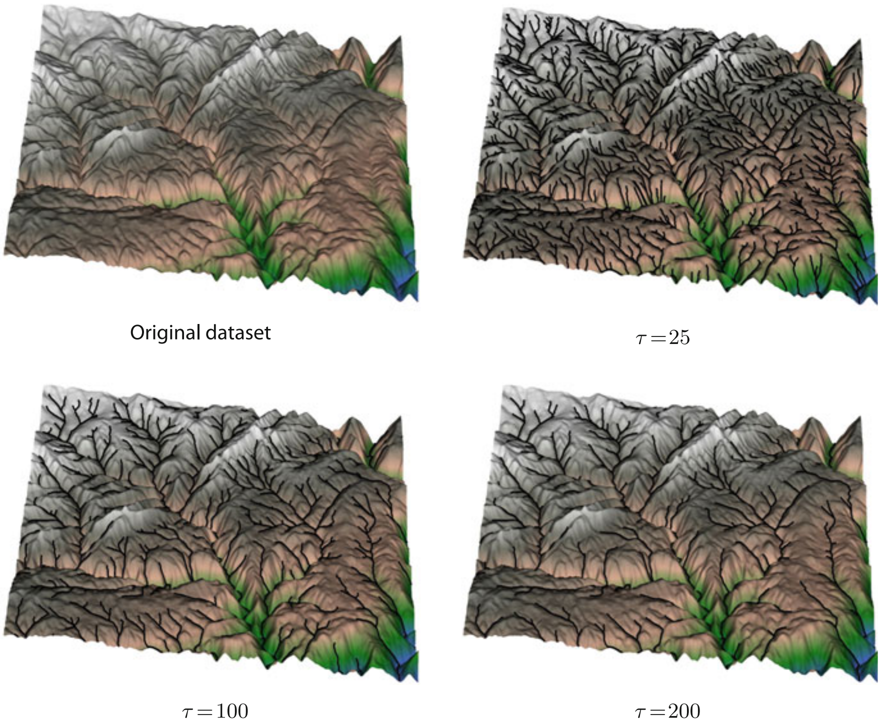
Local minima are known to exist in certain terrains, such as those consisting of Karst topography or natural basins (e.g. endorheic basins). However, most local minima present in terrain data are a result of the sampling procedure used to collect it, which may miss channels that are too small for the spacing of the grid and may pass between sample points (sometimes called “posts”). In addition, most data collection techniques cannot differentiate between tree cover, lake surfaces, and land, resulting in elevation inaccuracies and pits where there should be none. Collection procedures often fail to adhere to the Nyquist Sampling Theorem. One of the principal features of this new representation is that it facilitates the restriction of local minima, especially along the terrain’s channel network, by drawing a connection between the representation and the physics of the terrain’s formation.

Given our current reliance on digital map data, hand-held devices, and GPS navigation systems, the accuracy and compactness of terrain data representations are becoming increasingly important. Traditional representations do not maintain hydrographical information, critical when using algorithms to site dams or map flood plains, beyond what is present in the spatial data (elevations and grid locations). If more accurate data were to be universally available in a digital format traditional representations, which can limit the effectiveness of these algorithms, would be less necessary. For these reasons, it is imperative that a representation that retains important hydrographical properties while maintaining the characteristics of legal terrain be developed.

We present the *drill operator*, a mathematical operation that carves out terrain by “drilling” into it along the channel segments of its extracted channel network, changing its shape to fit the terrain’s profile at each position along the channel’s length. This representation allows for procedural generation of a terrain surface by mimicking the physical process of digging out the surface. We will demonstrate how this operator succeeds in the representation of legal terrains. We also provide the results of a series of accuracy tests on a real terrain dataset, providing evidence of the utility of the drill operator.

1.1 Existing Terrain Representations

The most common current representations of terrain are height fields (matrices of point elevations), and triangulated irregular networks (piecewise linear triangulated splines, known as TINs), both of which fail to capture the richness of the physics of the terrain surface. Height fields are two dimensional grids of elevation values, isometric to greyscale image data. Each grid space, or “pixel”, is single-valued (contains a single elevation). A TIN is a similar representation, but the surface is represented by triangles with single-valued vertices. An example of a height field can be seen in



Original dataset

$\tau = 25$

$\tau = 100$

$\tau = 200$

Fig. 1 Four depictions of a 400 × 400 mountainous dataset visualized as a height field of pixels. *Color* indicates elevation, where *white* is the highest elevation, followed by *brown*, then *green*, and finally *blue* indicates the lowest elevation. Below each data set is the threshold value used to extract the channel network depicted on the surface of each terrain. Notice the difference in the densities of the channel pixels as the threshold increases

Fig. 1, the mountainous dataset that will be used for testing and comparison for the remainder of the paper.

Both representations benefit from the assumption that terrain is at least C^0 continuous. However, this assumption is not always true. Discontinuities are apparent in real terrain (e.g. the Grand Canyon), but the information necessary to represent them is lost as the terrain is stored as a height field or TIN, as multiple vertices at a single grid space would be required.

Also, the accuracy of both height fields and TINs is subject to the resolution of the data points. Coarser resolutions have a profound and negative effect on the accuracy of the representation (Gao 1997). This limitation can be somewhat mitigated by allowing for variable resolution. This is possible for both representations (such as in Abdelguerfi et al. 1998; De Floriani and Puppo 1995; Bartholdi and Goldsman 2004; Velho et al. 1999), but even this solution is limited as each representation is inherently grid-based, and as such is limited in its precision.

A second option for representing terrain surfaces is the use of Fourier transforms, or other surface-shape modeling functions, such as B-Splines (Farin 2002, Faux and Pratt 1979). While these methods are more powerful when modeling terrain, as they allow local control over the shape of the terrain and provide a degree of local coherence (in that the elevation of a pixel corresponds in some way to the elevation of its neighbor), they still model a surface that is overly smooth (C^1, i.e. continuous and differentiable everywhere), and represent no physical process of terrain generation.

In addition to surface representations, there are popular volumetric terrain representations. Terrain can be stored as a voxel grid, allowing for multiple soil types. While this representation can, with a fine enough resolution, produce legal terrains, it is often not feasible to store terrains in this manner due to its substantial memory footprint. A layered height field (Benes and Forsbach 2001) combines several advantages of each of height fields and voxel grids. In this structure, the terrain is divided into a two dimensional grid, like a height field, but each grid space contains an array of heights. Modeling surface sculpting presents a challenge similar to terrain modeling, and so the two share some data representations, such as the "slab" data structure (Agarwala 1999), used for surface sculpting of volumetric models by converting the surface into a series of height fields layered over volumetric information.

Mathematical operators have been used to represent terrain surfaces in the past. This work most closely resembles the work presented by Franklin et al. (2006), where the scoop operator is introduced. Variations on the scoop operator perform similarly to our drill operator. Given a trajectory, the scoop operator digs out a portion of the terrain, and a terrain dataset can be represented with a series of scoops. The scoop shapes are determined by low order polynomials, each with its own advantages and disadvantages.

Terrains can be generated from scratch in most cases by either fractal generation or erosion simulation. Fractal terrain generation does not result in legal terrains, as it relies on controlled randomization of the elevations of height fields. Height field pixels store no neighborhood information, and fractal generation schemes introduce more randomness than is generally seen in terrain due to a lack of user control and correlation to terrain features. Erosion simulations generate realistic looking terrains, but also rely on the underlying height field structure, and are not feasibly stored as a procedural generation. We seek a representation of terrain that produces legal terrains while being closely tied with physical processes that produced the terrain, such as digging, or erosion.

Various compression schemes are used to store terrain data. The most common are variations on the JPEG algorithm which are, in general, very successful. However, terrain surfaces have also been modeled using overdetermined linear systems (ODETLAP) as a training model to find a well-compressed lossy representation (Inanc 2008). These lossy compressions are comparable (and, in some cases, even exceed) JPEG compressions, though without regard to hydrographic data and physical processes we wish to retain with our representation.

2 Representing Terrain as a Series of Drill Operations

We represent terrain data as a series of mathematical "drill" operations applied to an initial high plateau. The series of operations are extracted from a given terrain, T, where each elevation in the $n \times n$ grid is referred to as a pixel $\mathbf{p}$. A list of operations can be stored in lieu of the height field. The process for determining the series of drill operations is as follows:

1. Extract the channel network of T, using any common method.
2. For each pixel in the channel network $\mathbf{p}_i$, collect cross sections of the channel centered at $\mathbf{p}_i$.
3. Find the union of all cross sections to determine the "thinnest" channel cross-section.
4. Use least-squares fitting to fit a quadratic function to the union.

The coefficients of the fitted quadratic function and the position of the pixel completely describe a single drill operation. Like surface shape-modeling functions, the drill operator maintains the local coherence of terrain through its connection to the process of digging, by mimicking the physical process of digging in a controlled and efficient way. The drill operator introduces local continuities, but does not prevent discontinuities, such as on the edge of a channel, where one might expect to see them.

2.1 Channel Network Extraction

To extract the hydrographic channel network from the terrain, we use a hybrid of the method first presented by O'Callaghan and Mark (1984) and the method presented by Metz et al. (2011). For each pixel of the terrain, prioritized based on elevation (lowest elevation with highest priority), the direction of the neighbor (of eight possible neighbors) with the highest priority is set as the flow direction. Once the flow directions have been determined, the pixels are sorted by priority, and flow accumulation is calculated, where each pixel contributes its flow value to that of its flow neighbor, cumulatively. The channel network consists of those pixels whose flow values exceed a user-defined threshold, τ. Slight variations in geometry or threshold value can have a profound impact on the shape and size of the channel network. In fact, determining the ideal threshold from which to extract the channel network is a closely related area of research, and as such τ is one of three user-defined parameters to our system. It is important to note that this is a black box process, and any channel network extraction tool can be used. It is also important to note that a pixel's entire flow is applied to a single neighbor, and so channels can not split. An example of an extracted channel network is seen in Fig. 1.

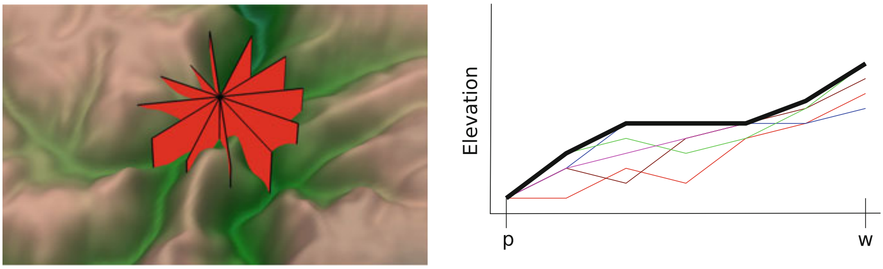


Fig. 2 A visualization of the process of determining the union of the cross sections of the terrain's channels. The *left* image depicts the process of gathering the cross sections. Each *red* plane is a cutting plane, and the cross section of the terrain determined by each one is collected. The *right* image depicts finding the union of all cross sections. The *colored thin lines* represent the set of all cross sections at pixel $\mathbf{p}$, each one a collection of elevations of length w. The *thick black line* represents the final union

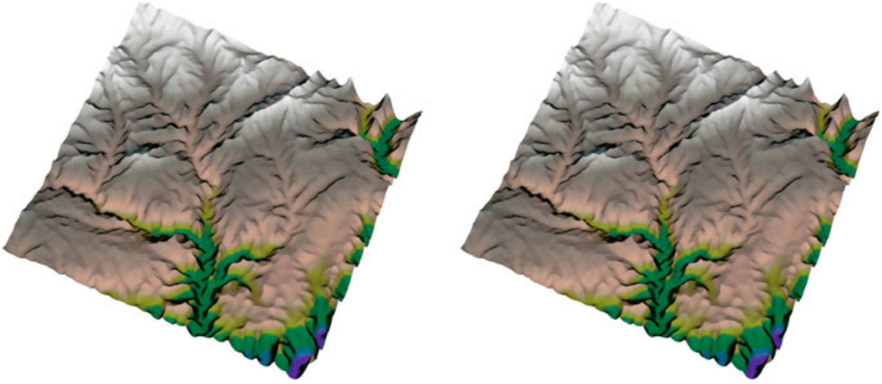
2.2 Determining Drill Shape

Next, the shape of the drill is determined for each pixel $\mathbf{p}_i$ in the channel network. To accomplish this, we fit a quadratic curve to the channel profile at $\mathbf{p}_i$, which then represents the shape of the widest drill that can fit in the channel. The overall channel profile is represented by the union of all collected cross sections at $\mathbf{p}_i$.

At each pixel $\mathbf{p}_i$, a set of 120 cross sections of the terrain is collected, each three degrees apart. This provides a profile of a channel at 120 distinct, uniformly spaced angles. A sample of this procedure is shown in Fig. 2. The cross sections span a distance w (a user-defined parameter) from the center of $\mathbf{p}_i$, the second user-defined parameter to our system. The larger the value for w is, the larger the area of influence on the terrain surface that is considered when determining the drill shape. For each pixel in the channel network, uniformly spaced sample elevation points are collected in each direction along the crossing plane representing the cross section. Most of these sample points do not fall directly in the center of a pixel, and so we use bilinear interpolation to estimate the elevation of any sample point that does not exactly match a pixel center.

Once the set of cross sections is collected, we calculate its union. To do this, we take the maximum elevation value at each sampled distance from $\mathbf{p}_i$, as depicted in Fig. 2. This new channel profile represents the widest channel that a drill can be fit to in order to conservatively carve the channel. A thinner drill will not carve a wide enough channel, and a wider drill will carve too much terrain away and make the channel too large. We then fit a quadratic equation to the union by treating the fitting as a constrained linear least-squares problem over the w elevations of the union. We constrain only the center point of the union, the elevation of $\mathbf{p}_i$. The coefficients of this fitted quadratic equation represent the shape of the drill at $\mathbf{p}_i$.

The terrain is completely represented by the list of pixels in the channel network and their associated drill's quadratic coefficients.



$\tau = 25$, $w = 10$, $k = 20$

$\tau = 25$, $w = 5$, $k = 10$

Fig. 3 The *left* image depicts the regenerated terrain with the lowest RMSE value, and the *right* image depicts the regenerated terrain with the lowest AHD error. The parameters responsible for the terrain are listed below the images

2.3 Terrain Regeneration

To regenerate the terrain from the drill representation, we begin with a flat plateaued terrain of elevation m, where m is the maximum elevation of T. A new terrain is procedurally generated by iteratively applying operations to this initial plateau, T_0. The ith drill, which corresponds with a pixel $\mathbf{p}_i$ in the channel network of T, is represented by a $(2k + 1) \times (2k + 1)$ matrix D_i, where k is referred to as the drill's "influence". Essentially, this is how large of an area of the terrain each drill operation affects, and is the third and final user-defined parameter to the system. The width and length of D_i must be odd because it must have a center pixel.

In order to drill the terrain T_{i-1}, D_i is centered over pixel $\mathbf{p}_i$. The procedural generation is defined in Eq. 1:

$$T_i = \min (T_{i-1}, D_i) \tag{1}$$

where the min () function operates over the $2k + 1 \times 2k + 1$ area of D_i and outputs a new matrix of minimum values between the pixels of D_i and corresponding pixels of T_{i-1}. The resulting terrain has been drilled, and the process is repeated for the next operation. The results of this procedural generation can be seen in Fig. 3.

3 Accuracy and Efficiency Testing

Accuracy tests were performed to determine how closely the drill representation modeled the 400 × 400 mountainous dataset seen in Fig. 1. All tests were performed in Ubuntu 11.04 with a quad-core AMD Phenom II X4 945 Processor, with 8GB of RAM. The code was written in MATLAB.

The three parameters of the system, as described above, are the threshold used to extract the original terrain's channel network, τ, the size of the cross section, w, and the influence of a drill (size of the representative matrix), k. Each of 3 thresholds, 3 cross-section sizes, and 6 influence values were used to build regenerated terrains in this factorial experiment. For each set of parameter values, the total error between the generated terrain and the initial terrain was calculated. Two metrics were used: the standard root mean squares error (RMSE) metric, and the averaged Hausdorff distance (AHD) metric as described by Stuetzle et al. (2011).

RMSE measures the root of the average squared difference in heights across the terrain, as shown in Eq. 2:

$$RMSE\left(T_0, T_1\right) = \sqrt{\frac{\sum_{x<X} \sum_{y<Y} \left(T_0\left(x, y\right) - T_1\left(x, y\right)\right)^2}{X * Y}} \tag{2}$$

where each dataset, T_0 and T_1, is $X \times Y$ pixels, and $T\left(x, y\right)$ is the elevation value at location x, y.

For any two sets of pixels (in this case, the sets of all pixels in the channel network of each terrain), AHD finds the average of the shortest distance between their pixels, as shown in Eq. 3:

$$AHD = \max \left\{ \overline{\inf_{\mathbf{p}_j \in N_\tau^{T_1}, \mathbf{p}_i \in N_\tau^{T_0}} d\left(\mathbf{p}_i, \mathbf{p}_j\right)}, \overline{\inf_{\mathbf{p}_i \in N_\tau^{T_0}, \mathbf{p}_j \in N_\tau^{T_1}} d\left(\mathbf{p}_i, \mathbf{p}_j\right)} \right\} \tag{3}$$

where $\mathbf{p}_i \in N_\tau^{T_0}$ is the ith pixel of the set of channel network pixels of T_0 extracted using threshold τ, $N_\tau^{T_0}$. The overline means "mean value of". It is important to note that AHD is limited to the channel networks of the terrain, whereas RMSE is applied globally. Limiting RMSE to only N_τ would not give an accurate picture of how close the terrains' hydrography networks are, since the network pixels are found by looking at the global flow pattern. Even if the elevations of the pixels in N_τ are comparable, it does not indicate that the overall terrain shape is similar. In addition, slight variations in the location of the pixels in N_τ would render RMSE unusable. We look at the results for both metrics and analyze them in context of their maximum error and what they mean from a physical standpoint.

3.1 Results and Discussion

The results for our accuracy tests are provided in Fig. 4. The left column of Fig. 4 presents the data for the RMSE metric, and the data for the AHD is on the right. The error divisor for RMSE was the total elevation range of the original terrain, whereas the error divisor of AHD was the distance between pixel 0,0 and pixel n,n, the corners

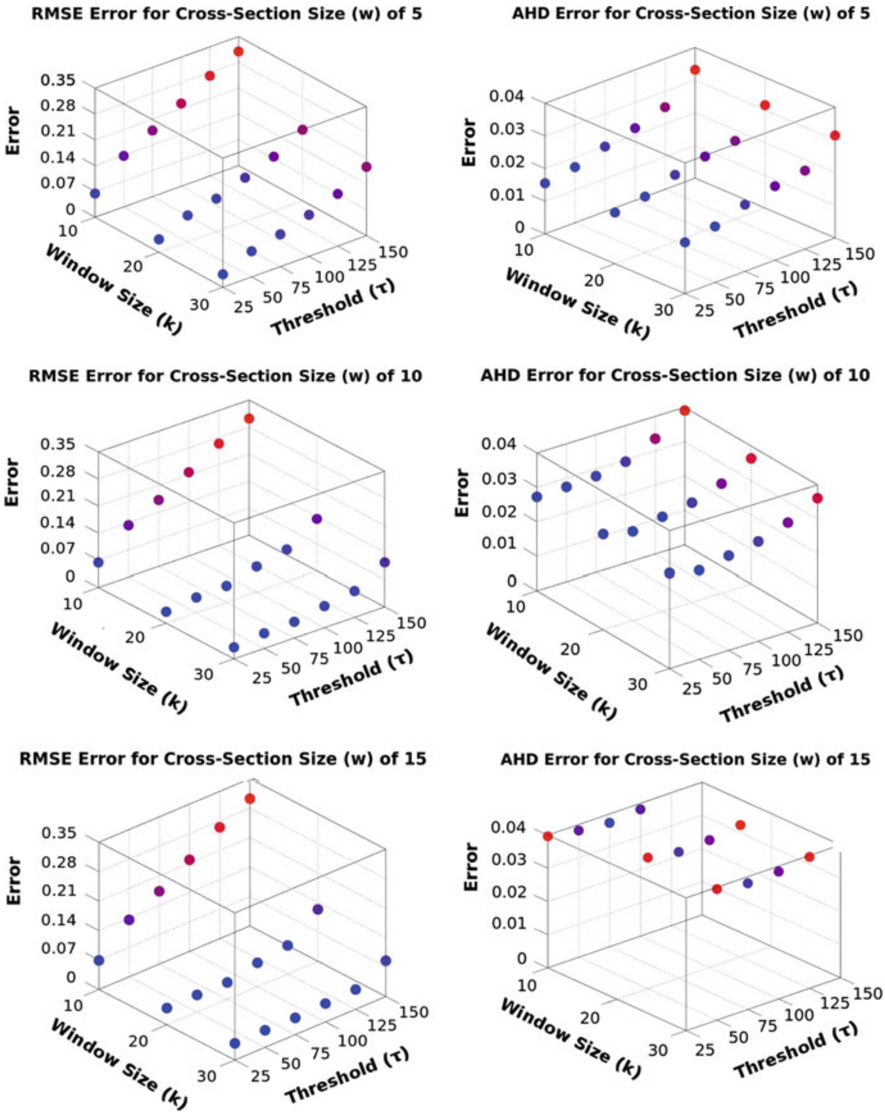


Fig. 4 The results of our accuracy testing. The *left column* depicts the error measured using the RMSE metric, and the *right column* depicts error measured used the AHD metric. Each *row* shows a different value for the cross section window size (parameter w). Each metric's data is normalized. *Red* indicates high error, whereas *blue* indicates low error, local to each dataset

of the terrain. Since elevation is ignored for this divisor, it is possible for AHD error be greater than 1.0.

τ values of 25 created considerably denser channel networks than τ values above 100, and so they are both more accurate and take considerably longer to calculate. The two channel networks can be seen in Fig. 1. For reference, the maximum flow value for a pixel in our dataset is 102,245, and a threshold of 25 resulted in a channel network with $\approx$19, 000 pixels, whereas a threshold of 100 resulted in a network of $\approx$12, 000 pixels. Calculating the drill representation of a terrain with a flow threshold of 25 took approximately eight minutes, and times decreased linearly with the increase in threshold value. The time for regenerating the terrain from the drill representation was quicker by a factor of ten, approximately. It is important to note that time optimization was not a focus for this paper, and there exist many techniques that will be utilized in the future to reduce the time the algorithm takes.

There are several observations to be made regarding our accuracy data. The first is that, with "good" parameters (meaning none that create outliers in the error data), the accuracy is actually quite high with regard to both RMSE and AHD metrics, between 0.015 and 0.025. As with any algorithm, there is a trade-off between time and accuracy. The lower the threshold, the more pixels in the hydrography network, and so the slower the process, but the more accurate the results become. This is true in every case, and it makes intuitive sense.

τ is the most influential parameter with regard to both metrics. This makes sense, since the drill shape calculation is constrained to pass through the center of N^i_τ. Since AHD measures how close the channel networks are to one another, the more pixels there are in the channels, the more likely it is that the channel networks overlap. By nature of the AHD, this will result in very low error. If the channel network is not dense enough, then the drill may not reach much of the terrain at all, and so the RMSE is also very much influenced by the value of τ. More interesting is that the minimum RMSE occurred for when $w = 10$ and $k = 20$. It makes sense that a drill will need some influence over the area around the channel in order to reduce the RMSE, because without it there would be large areas of the terrain untouched by any drill, adding considerably to the overall error (that AHD would ignore).

From a purely visual standpoint, many of the terrains pass the eyeball test. A side-by-side comparison of the terrains that represent the lowest error for each of the metrics is found in Fig. 3. Notice is that the channels do seem to be somewhat wider in the RMSE winner, whereas they are more pointed, but areas of the terrain are missed by drills completely, in the AHD winner. Interestingly, the left image in Fig. 3 is one of the terrains we deem to be the most "visually pleasing". These often have higher k and w values, which may be a result of drills that flatten artifacts out of the terrain. This result demonstrates why, when choosing parameters for the drill operator, it is often smart to take both error metrics into account.

4 Conclusion

We have presented the drill operator, the first in a series of mathematical operators applied to terrain datasets. It creates a robust, effective, and accurate representation of terrain data. It adheres strictly to two of the three criteria laid out in Sect. 1 for legal terrain generation. First, it allows for the existence of discontinuities, which can exist at the edge of, and on the border between, two drills. Second, its conception and execution is based on an actual physical process that can affect terrain shape, and that information is inherent in the representation. However, while locally the drill prevents minima due to the monotonically increasing nature of the drill shape, it has no such restriction on a global scale in its present state. This is discussed further in Sect. 4.1.

The drill has been shown to represent terrain to a sufficient degree of accuracy, and its resultant terrains are visibly pleasing and comparable to the original terrain. Hydrography information is well maintained, and as such our representation holds more information about the terrain, especially regarding its formation and hydrography, than traditional spatial representations. This representation will extend naturally to compression schemes, and work well in algorithms regarding path planning, flood plain mapping, and dam siting, as they all require hydrographically accurate terrain. In addition, the representation has a naturally deep and interesting mathematical component that will be explored and exploited in the future.

4.1 Future Work

A single drill is represented by a pixel and a polynomial, a total of 4 values that must be stored. These can be represented as integers. To represent a 400 × 400 terrain well enough, approximately 13,000 drills are required, bringing the total number of values stored for a single terrain to 52,000, or roughly one third of the total number of values stored in a height field. This yields an RMSE of less than 4 %, and is visually pleasing. While this does not represent a compression scheme in its current form, with further emphasis on optimization (both spatial and temporal) this representation will be a viable candidate for compression. In addition, it more accurately maintains hydrography and formation information, an advantage over other more frequently used schemes.

There are several ways in which the algorithm can be optimized. The first is by storing channel segments as opposed to individual pixels, such as with Freeman chain codes (Freeman 1974) or line generalization (Ramer 1972). If a channel network can be stored as a series of segments, with begin points, end points, and channel trajectories, along with a function of drill shape (allowing it to change along the length of the channel), then the overall storage of a terrain can be greatly reduced. Storing the channels as trajectories also allows for a post-processing step that forces the trajectories to be monotonic and decreasing, thus forcing a lack of local minima and allowing the drill operator to adhere to the second criterion for a “legal terrain” described in Sect. 1.

With regards to temporal optimization, the drill creation algorithm bottlenecks at the quadratic function fitting. A better method of function fitting (or an easier-to-fit shape) may significantly improve the performance of the algorithm. Drills can be any shape, including simple cones, and have a width along the bottom. If the restriction that the drill shape functions be monotonic and increasing is lifted, a drill can be ball-shaped, which could model caves. These new parameters would allow for more customizability while maintaining a small memory footprint.

Two more operators will be added to the set of operators: the erosion operator, and the mudslide operator. The erosion operator picks up sediment from a high elevation and deposits it at a lower elevation, and the mudslide operator adds sediment to the terrain along a trajectory. Deeper mathematics of these three operators will be explored, including how to define the composition of two operations, and whether that creates a group over the composition operation.

Finally, we will investigate ways to post-process terrain to deal with small areas untouched by drill operations. A simple interpolation of the untouched areas may be a solution, or more sophisticated algorithms may be more useful. Also, the system, as it currently stands, has three user-defined parameters (τ, w, and k). We will investigate ways to automatically determine the ideal value for each of these parameters before operation extraction, thus not relying on user knowledge to create the best representation. Ideally, every operator will store its own value for k.

Acknowledgments This research was partially supported by NSF grants CMMI-0835762 and IIS-1117277. We would like to thank the reviewers for their helpful comments.

References

Abdelguerfi M, Wynne C, Cooper E, Roy L (1998) Representation of 3-d elevation in terrain databases using hierarchical triangulated irregular networks: a comparative analysis. Int J Geogr Inf Sci 12(8):853–873. http://www.tandfonline.com/doi/abs/10.1080/136588198241536

Agarwala A (1999) Volumetric surface sculpting. Ph.D. thesis, Massachusetts Institute of Technology

Bartholdi JJ, Goldsman P (2004) Multiresolution indexing of triangulated irregular networks. IEEE Trans Visual Comput Graphics 10(4): 484–495

Benes B, Forsbach R (2001) Layered data representation for visual simulation of terrain erosion. In: Proceedings of the 17th spring conference on computer graphics, Washington

De Floriani L, Puppo E (1995) Hierarchical triangulation for multiresolution surface description. ACM Trans Graph 14:363–411

Farin G (2002) Curves and surfaces for CAGD: a practical guide, 5th edn. Morgan Kaufmann Publishers, San Francisco

Faux ID, Pratt MJ (1979) Computational geometry for design and manufacture. Halsted Press, New York

Franklin WR, Inanc M, Xie Z (2006) Two novel surface representation techniques. In: Autocarto 2006. Cartography and geographic information society, Vancouver, Washington, (25–28 June 2006)

Freeman H (1974) Computer processing of line-drawing images. ACM Comput Surv 6(1): 57–97. http://doi.acm.org/10.1145/356625.356627

Gao J (1997) Resolution and accuracy of terrain representation by grid dems at a micro-scale. Int J Geogr Inf Sci 11(2):199–212

Inanc M (2008) Compressing terrain elevation datasets. Ph.D. thesis, Rensselaer Polytechnic Institute

Metz M, Mitasova H, Harmon RS (2011) Efficient extraction of drainage networks from massive, radar-based elevation models with least cost path search. Hydrol Earth Syst Sci 15(2):667–678

O'Callaghan JF, Mark DM (1984) The extraction of drainage networks from digital elevation data. Comput Vis Graphics Image Process 28:323–344

Ramer U (1972) An iterative procedure for the polygonal approximation of plane curves. Comput Graphics Image Process 1(3):244–256http://dx.doi.org/10.1016/S0146-664X(72)80017-0

Stuetzle C, Cutler B, Chen Z, Franklin WR, Kamalzare M, Zimmie T (2011) Ph.d. showcase: measuring terrain distances through extracted channel networks. In: 19th ACM SIGSPATIAL International conference on advances in geographic information systems, Chicago, 2011

Velho L, de Figueiredo LH, Gomes J (1999) Hierarchical generalized triangle strips. Vis Comput 15:21–35. doi:10.1007/s003710050160

A New Algorithm for 3D Isovists

Wassim Suleiman, Thierry Joliveau and Eric Favier

Abstract Isovist or vision field computing is an interesting topic with many applications in different fields: security, wireless network design, or landscape management. In all existing solutions, a 3D environment appears to be the most challenging task and few solutions exist for detecting the obstacles that limit the vision field. In this paper a new algorithm is presented for isovist calculation that can detect all objects, which block the sight in a 2D and 3D environment. Then, a demonstration with GIS data is given and some visibility indices are also presented.

Keywords Field of vision · GIS · Photometry · Isovist · Information visualization · Ray tracing · Virtual reality · Visualization techniques and methodologies · Space syntax

1 Introduction

The Isovist is defined as the area that can be seen from a given point in space (Benedict 1979). Another definition is introduced by Gibson (1983), Conroy Dalton and Bafna (1983), which is based on the aggregation of all lines of sight as initialized from the

W. Suleiman (✉) · T. Joliveau
ISTHME-EVS CNRS UMR 5600, Université Jean Monnet-Saint-Etienne,
Saint-Etienne, France
e-mail: wassim.suleiman@univ-st-etienne.fr

T. Joliveau
e-mail: thierry.joliveau@univ-st-etienne.fr

E. Favier
University of Lyon-Ecole nationale d'ingénieurs de Saint-Etienne DIPI (ENISE),
Saint-Etienne, France
e-mail: eric.favier@enise.fr

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_11,

observer point. Vision field computation emerged as a computing question from the need to address different kinds of problems: where to position the watchman in a museum or the surveillance cameras in a bank? Or more recently, how to design a wireless network, which covers a large campus at a minimum cost? How to dispose buildings in a development area to create a pleasant environment, where people do not feel enclosed, and can have a pleasant view over natural landscapes like mountains, lakes or parks? Where is the best place in a city for a publicity panel or a shop in terms of visibility?

Much effort was put into understanding the relationship between built form and perceived space. Visual perception of space and the relation between human activities and the environment is studied in Ittelson (1960). Other studies showed that people prefer to pause and linger on in the 'inside corners' of spaces (Ashihara 1984). Some researchers used 'viewshed' for analyzing visual effects of terrain maps (Lynch 1976). Later, 'Space syntax' was presented to describe and analyze the character of space and predict human behavior in space (Hillier and Hanson 1984).

Our goal is to propose a new algorithm to calculate a 3D Isovist in an urban environment based on the natural ground surface represented by a digital evaluation model (DEM).

This paper is organized as follows: Sect. 2 presents the related work about isovists in 2D/3D environments, current solutions and programs. Section 3 explains a method for calculating intervisibility in a 3D vector environment. Section 4 focuses on our research on a new Isovist algorithm for 2D environments. Section 5 generalizes our researches for a new Isovist algorithm within 3D environments. Section 6 presents our results by applying our method to a 3D GIS data model of an area of Saint-Etienne in France. In the last section, a conclusion of our work is given.

2 Related Work

2.1 2D Isovist Programs

Software that computes 2D Isovists and Space syntax indices is presented in Do (1997). Many techniques are used: Ray Tracing (Tcl-Light), shadow casting (Do 1994a,b), removing occluded walls in the AutoCAD environment using AutoLisp (Do 1994a,b) and then finding visible walls (Do 1995). A ray tracing algorithm is used to develop a tool based on Arcview[1] (Rana 1994). Visual permeability is calculated using isovist and occlusion maps (Christenson 2010). Other methods were developed like walking agent (Batty and Jiang 2010) and intervisibility (Turner et al. 2001).

[1] ArcView 3 version 'ESRI'

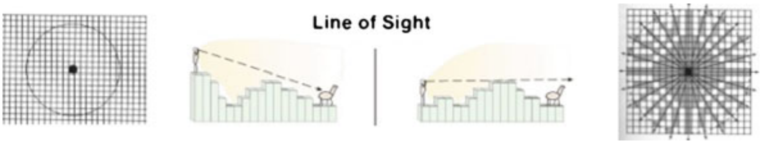


Fig. 1 Visibility in raster mode with regular pixelisation (Source Brossard and Wieber)

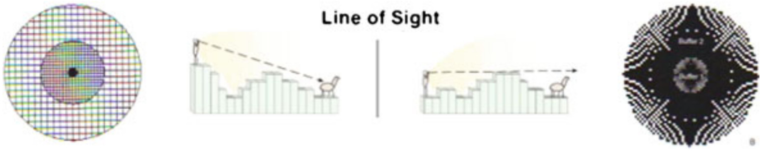


Fig. 2 Visibility in raster mode with variable pixelisation. Source Brossard and Wieber

2.2 3D Isovist Programs

The first step of computing visibility in a 3D environment was to build a 2.5D environment by adding the height of the buildings to a Digital Evaluation Model (DEM), which describes a topographic terrain. Visibility could then be calculated in the raster mode using lines of sight and ray tracing (Lake et al. 2000; Brossard et al. 2008; Franklin and Ray 1994; De Floriani and Magillo 1994; Van Kreveld 1996; Floriani and Magillo 2003; Fishman et al. 2009; Pyysalo et al. 2009; Morello and Ratti 2009) Fig. 1.

A regular grid is used to create a raster image and a ray tracing algorithm is applied to test the intervisibility between the vantage point and the other pixels. As it is shown in Fig. 1, a high resolution image requires considerable calculation time while a low resolution image will lead to imprecise results. A later idea was to use the Multiresolution concept. We use a high resolution in the area near the vantage point and a lower resolution in the other parts of the image Fig. 2.

In this raster method, the precise shape of the building is lost because of pixelisation and the identification of the buildings becomes impossible.

Other researchers followed a 3D-isovist-like approach (Pyysalo et al. 2009; Morello and Ratti 2009) Fig. 3. The basic idea was to use a 3D raster voxel model; a voxel is a volume element, representing a value on a regular grid in three dimensional space. Adaptive transparency is used to add buildings and vegetation to the terrain and calculate the 3D Isovist.

A program is developed with Matlab to calculate the Isovist by using the ray tracing algorithm (Bilsen 2009). The program works in a 3D environment with a plane terrain model. Later, a program is used in Bilsen (2010) to calculate Space Syntax indices within the 3D environment.

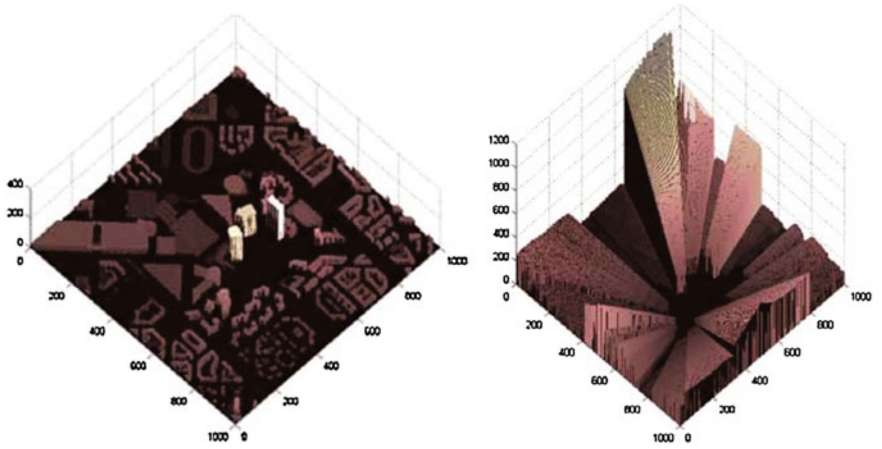


Fig. 3 Voxel model (Morello and Ratti 2009)

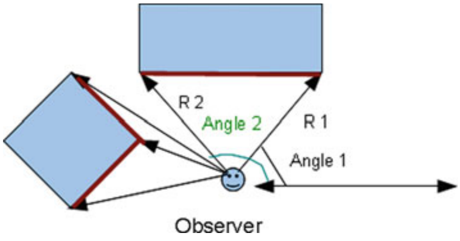


Fig. 4 Angle of vision associated with the segments which limit the vision

This 3D Isovist calculation is based on the ray tracing idea to scan the complete space. To achieve a satisfying level of precision of the 3D isovist, it is necessary to execute a complicated calculation process ($360 * 180 = 64800$ tracing rays).

3 New Algorithm for 2D Isovist

Our algorithm for a 2D environment is based on the assumption that "2D space is an aggregation of segments". In the algorithm we are looking for segments, which block the sight of vision Fig. 4.
We create a list of segments:

$$S = [\text{Segment}_1,\ \text{Segment}_2, \ldots, \text{Segment}_\text{N}]$$

For each segment (Segment$_\text{i}$) we have two extremities ($[(\theta 1_i, r1_i)$; $(\theta 2_i, r2_i)]$) with the polar coordinates centered at the observer point.

Definition 1: The angle of vision for $Segment_I$ is the angular interval, $AVS_I = [\theta 1_I, \theta 2_I]$. The module of the angle of vision is:
$|AVS_I| = |\theta 1_I - \theta 2_I|$.
Definition 2: The free segment is a segment which has two visible ends for the observer.
Definition 3: The free vision field is an angle of vision where there is no obstacle to block the sight of light.

FVF is Free Vision Field
BVF is Blocked Vision Field
S is List of segment
F is the free segment list
$|AVS_I|$ is the value of the angle of vision.

```
                    2D Isovist algorithm
BEGIN
Initialize FVF = [0,2π] , BVF= ∅
While (S≠ ∅ ) do:
    Search for the free segments in S, and then order them with
    the increasing$|AVS_I|$. cf. Fig. 6
    b  F = [$Seg_c, Seg_v, Seg_t, \dots\dots$] .
    While (F ≠ ∅ )do:
        ($Seg_c$) blocks the visibility in the angles
        $A_c$ ={[θ1c, θ2c] ∩FVF}
        BVF = BVF ∪ $A_c$
        S = S – $Seg_c$ , F=F-$Seg_c$
        FVF = FVF - $A_c$.
    ENDWHILE
    cf. Fig. 6 c  Eliminate all segments who have angle of vision
    out of the free vision field from S. (i.e., the intersection be-
    tween their angle of vision and the free vision field is emp-
    ty).
ENDWHILE
RETURN Results = FVF ∪ BVF.
END.
```

The result of this algorithm is a list of angular intervals. Each angular interval is associated with the segment which blocks the vision or with the free space (free vision interval) Fig. 5.
This means that the isovist can be considered an aggregation of items in the following form [0, 2π]* {Segments}.

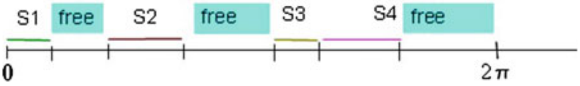


Fig. 5 Isovist for 2D environment

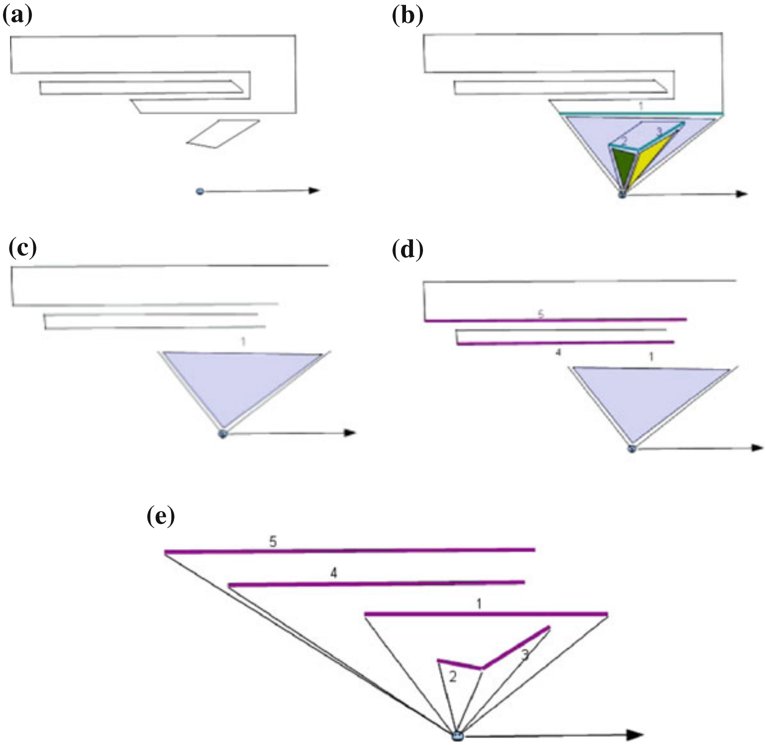
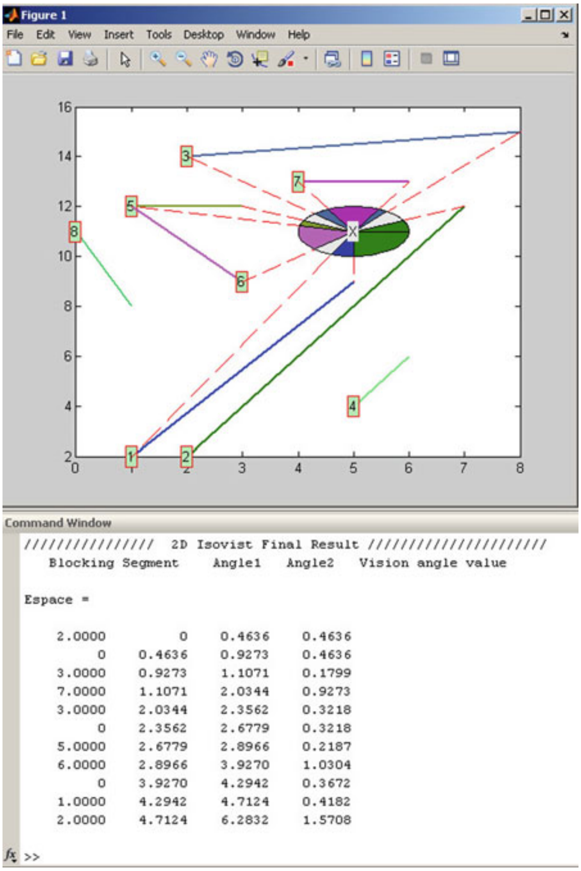


Fig. 6 2D Isovist **(a)** start **(b)** finding the free segments **(c)** eliminating the hidden segments **(d)** finding the free segments **(e)** results

The algorithm identifies all segments that limit the vision field. A problem is to find the segments which block the vision for each end of the segments list S. In this example N segments form the polygons so we have N extremities, the complexity of this algorithm will be of order N^2 = (segments number)^2 calculation of segment intersections. This performance is ameliorated by applying ≪ Sweep line algorithm ≫ (Bentley and Ottmann 1979) which had a complexity order of N*log(N).
By connecting the segments (or the segments parts) which block the vision field, we will obtain a visibility polygon as mentioned in Sect. 2.2. Figure 6 shows an example of the algorithm in a 2D environment.



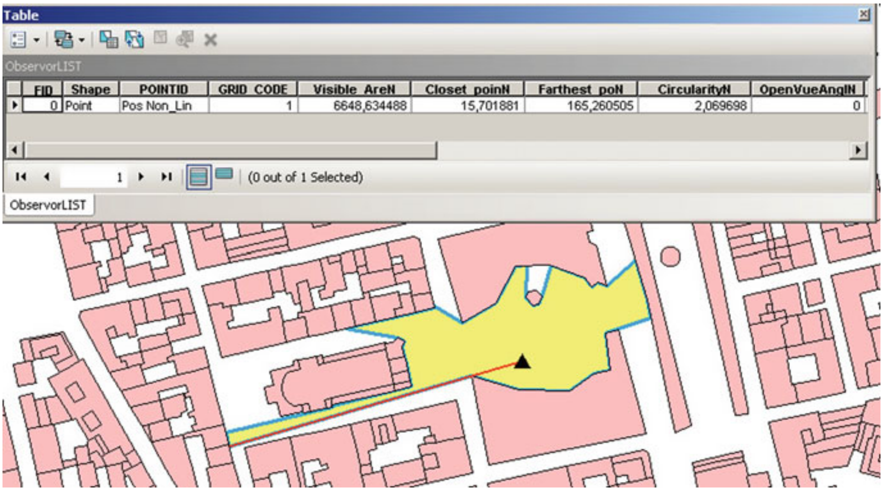
```
//////////////// 2D Isovist Final Result //////////////////////
   Blocking Segment    Angle1   Angle2   Vision angle value

Espace =

    2.0000         0    0.4636    0.4636
         0    0.4636    0.9273    0.4636
    3.0000    0.9273    1.1071    0.1799
    7.0000    1.1071    2.0344    0.9273
    3.0000    2.0344    2.3562    0.3218
         0    2.3562    2.6779    0.3218
    5.0000    2.6779    2.8966    0.2187
    6.0000    2.8966    3.9270    1.0304
         0    3.9270    4.2942    0.3672
    1.0000    4.2942    4.7124    0.4182
    2.0000    4.7124    6.2832    1.5708
```

Fig. 7 2D Isovist results: first colon is the segment number, second line the first angle, third colon the second angle and last line is the vision angle

The result of this example is:

$[0, \theta 1_1]$ Free field.
$[\theta 1_1, \theta 1_3]$ Blocked by (Segment_1).
$[\theta 1_3, \theta 2_3]$ Blocked by (Segment_3).
$[\theta 1_2, \theta 2_2]$Blocked by (Segment_2).
$[\theta 2_2, \theta 2_1]$Blocked by (Segment_1).
$[\theta 2_1, \theta 2_4]$Blocked by (Segment_4).
$[\theta 2_4, \theta 2_5]$ Blocked by (Segment_5).
$[\theta 2_5, 2\pi]$ Free field.

Figure 7 presents an implementation of our Isovist method in a 2D virtual environment. The method used here is similar to the output point of view in (Do 1997), but the difference lies in the methodology of calculation, what makes it possible to extend the principle to a 3D environment while (Do 1997) is limited to 2D environments.



	FID	Shape	POINTID	GRID_CODE	Visible_AreN	Closet_poinN	Farthest_poN	CircularityN	OpenVueAngIN
▸	0	Point	Pos Non_Lin	1	6648,634488	15,701881	165,260505	2,069698	0

Fig. 8 Isovist 2D applied on GIS buildings databases and exported to arcgis environment

Figure 8 present the results of our method as applied on the GIS 2D databases with the extraction of the visibility polygon with some morphometry measurements (Rana 1994), in Fig. 8 we calculate the visible area, the distance to the closest point, the distance to the farthest point, the circularity index, the open view angle and the farthest point direction.

This algorithm makes it possible for us to calculate more parameters related to the objects visible for the observer. Figure 9 presents the 2D isovist calculation in a built commercial environment in Saint Etienne city center. Regarding our method, the dominant type of shops visible for the observer is “Equipement de la personne” with percentage of 61 %, and the open view is 5 %. For the specific view angle presented in Fig. 9, the dominant type of shops visible is “Alimentaire” with 17 %, and the dominant type of buildings is “Résidence” with 54 %.

4 A New Algorithm for Computing a 3D Isovist

The algorithm used to calculate the isovist in a 3D environment is based on the same kind of assumption as presented before: “all surfaces in the 3D environment are an aggregation of polygons”.
A point with spherical coordinates is centered at the observer point.

Figure 10 is defined by (r, ϕ, θ) with $r \in \mathbb{R}$, $\phi \in [0, 2\pi]$, $\theta \in [0, \pi]$, , a polygon surface is a list of ordered points $\left\{p1_i, p2_i, \ldots, pn_i\right\}$ defined as a series of triples

$$\{(r1_i, \phi 1_i, \theta 1_i)\,, (r2_i, \phi 2_i, \theta 2_i)\,, \ldots, (rn_i, \phi\, n_i, \theta\, n_i)\}$$

By projecting this on the unit sphere we obtain the points:$\{(1, \phi 1_i, \theta 1_i), (1, \phi 2_i, \theta 2_i), \ldots, (1, \phi\, n_i, \theta\, n_i)\}$, connected by the spherical arcs:$\{(1, \phi 1_i, \theta 1_i), (1, \phi 2_i, \theta 2_i)\}\,; \ldots \{(1, \phi(n-1)_i, \theta(n-1)_i), (1, \phi\, n_i, \theta\, n_i)\}$ Fig. 11.

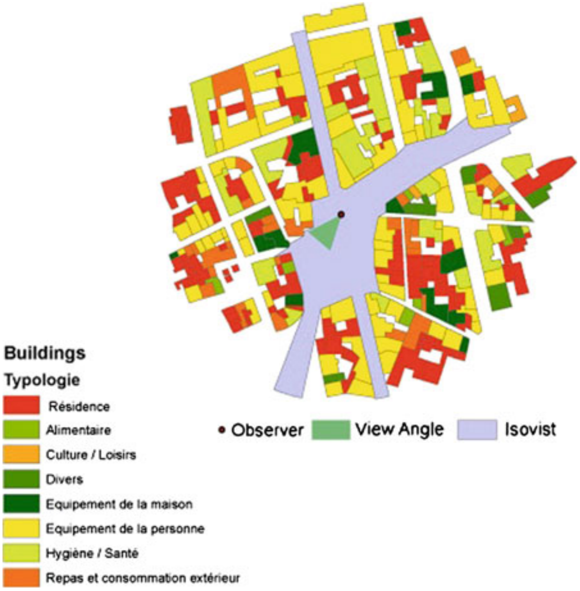


Fig. 9 2D Isovist with building type and view angle

Fig. 10 Spherical coordinate

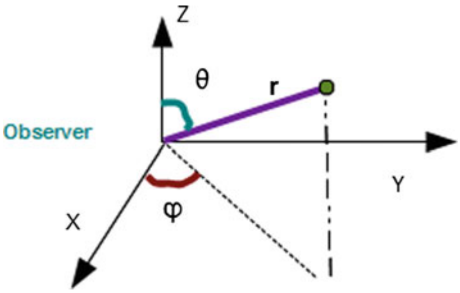
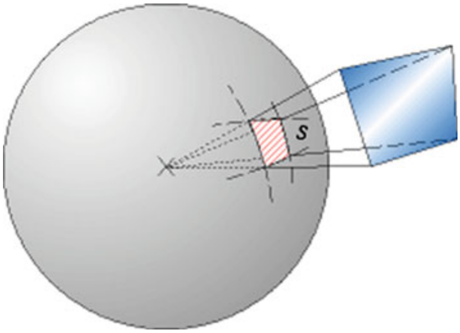


Fig. 11 The projection on unit sphere (Wikipedia)



Because all radiuses are equal to 1 this projection can be defined as a series of tuples:$\{(\phi 1_{i}, \theta 1_{i}), (\phi 2_{i}, \theta 2_{i}), \ldots, (\phi n_{i}, \theta n_{i})\}$.
The surface created by the projection is the solid angle of the polygon visibility.

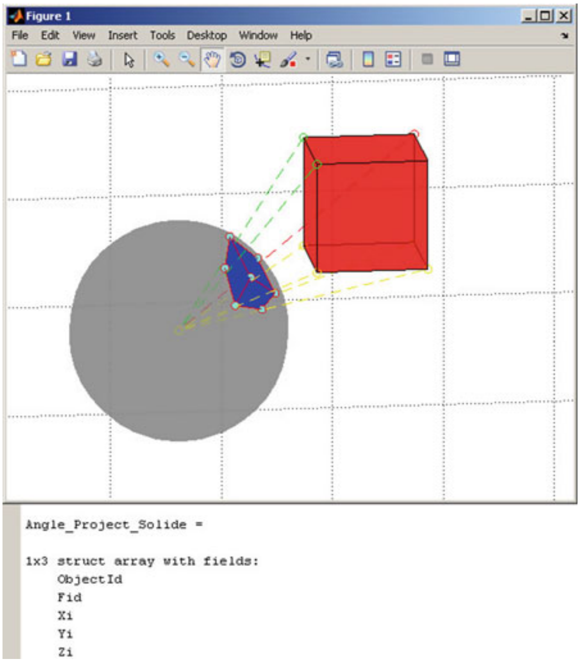
Definition 4: $SAVP_I$ (The solid angle of vision for $Polygon_I$) is the solid angle at which the observer can view $Polygon_I$ in it. $|SAVP_I|$ is the module of the angle of vision. Which is the surface area enclosed by $SAVP_I$ on the unit SPHERE.
Definition 5: The free polygon is a polygon whose edges (or the enclosed frontier segments) are visible from the observer.
Definition 6: The free vision field is a solid angle of vision where no obstacle can block the sight of light through it.

FS is the free surface on the unit sphere.
BS is blocked surface on the unit sphere
P is the list of polygons
F is the list of the free polygons
$|SAVP_I|$ Is the module of the solid angle of vision for $Polygon_I$

3D ISOVIST algorithm

```
BEGIN
Initializing FS= {the entire unit sphere}, BS = $\emptyset$ .
While (P $\neq \emptyset$) do :
    Search for the free polygons. Order them with the increasing
    $|SAVP_I|$, $F = [Polygon_c, Polygon_v, Polygon_t, \ldots\ldots]$.
    While (F $\neq \emptyset$ ) do:
        ($Polygon_c$) blocks the visibility in the solid
        gles $SA_c = SAVP_c \cap FS$.
        $BS = BS \cup SA_c$
        $P = P - Polygon_c$ , $F = F - Polygon_c$.
        $FS=FS - SA_c$ .
    ENDWHILE
    Eliminate all polygons that have an angle of vision outside the
    free vision field (the intersection between their solid angle of vi-
    sion and the free vision field is empty) from P.
ENDWHILE
RETURN Results = $FS \cup BS$
END.
```

This algorithm aggregates surfaces on the unit sphere (solid angles), associated with the polygons identifiers.



```
Angle_Project_Solide =

1x3 struct array with fields:
    ObjectId
    Fid
    Xi
    Yi
    Zi
```

Fig. 12 3D Isovist implementation with Matlab. The output is a series of solid angle with the object identification and facade identification

The algorithm tests for the edges of each polygon whether they are visible from the observer point. So its complexity is of order P = (polygon number) segment-polygon intersection.

12 presents the implementation of our 3D isovist with 3D virtual cubes.

5 Application to 3D Environment

Some tests have been realized with 3D buildings located on a hilly area in Saint-Etienne (France). The test uses 3D GIS data for the buildings and a DEM (Digital Elevation Model) of 1 m resolution for the terrain. Both building and terrain data were transformed into a set of polygons as facades for buildings and triangles produced of a Delaunay triangulation for the terrain (Suleiman et al. 2011). Our algorithm calculates the 3D Isovist from a vantage point located above the ground. The code was written by using Matlab mapping tools.

The area's radius measures 500 m and the model contains 17 buildings (231 facades). The execution time was 1.5 min on an AMD Athlon 30 Dual core 4800+. The results are illustrated as a set of solid angles with their corresponding polygons

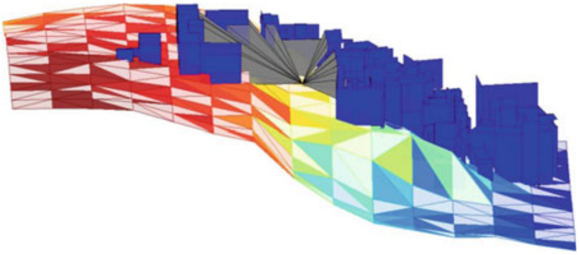


Fig. 13 3D ISOVIST with GIS data. The building facades visible from a vantage point

(building faced or triangle tessellation) which block the vision in that direction Fig. 13.

5.1 Combining the 3D Isovist with a Picture

To view the results in a more realistic way, it is possible to project the spherical surface related to our 3D Isovist on a rectangular plane of a calibrated photo model. The result is an image of the 3D environment taken from the vantage point with identified direction and view angle. We can observe the similarity between the real and the calculated images Fig. 14.

To realize the match of Fig. 14 between the modeled object image and the real one, we make a fit between the primitives (corners, edges) of these two images. This operation is done due the tuning of the initial position and direction of the camera model. Automatic adjustment was introduced by many techniques like Hough transform (Boehm et al. 2002), object tracking (Rosenberg and Werman 1998), Simultaneous Pose and Correspondence (David et al. 2004), (Sourimant et al. 2009), Hausdorff distance (Zhao et al. 2005), Lie Algebra (Ortegón-Aguilar and Bayro-Corrochano 2006) and the tracking of complex structures (Drummond and Cipolla 2000, 2002). These methods essentially depend on the accuracy of the model and on the assumption that one object exists that could be isolated in the image with no extern occlusion. In our test we did not use this kind of algorithm because there are many objects hidden behind each other. Some mobile elements like cars or fixed ones like trees are not modeled in the GIS database and can hide some parts of the building facades we attempt to visualize. Actually theses methods are still limited to simple non-occluded building cases. Our tests in Fig. 14 were made by taking an image using a calibrated camera from a given place and thanks to the GPS tool and accelerometer, we receive the initial position and direction of the camera image; we use our 3D isovist with the initial position and direction and the camera model to calculate the image in the 3D environment. Then, we made a manual correction of the preliminary position and direction of camera model, in order to make the primitives fit.

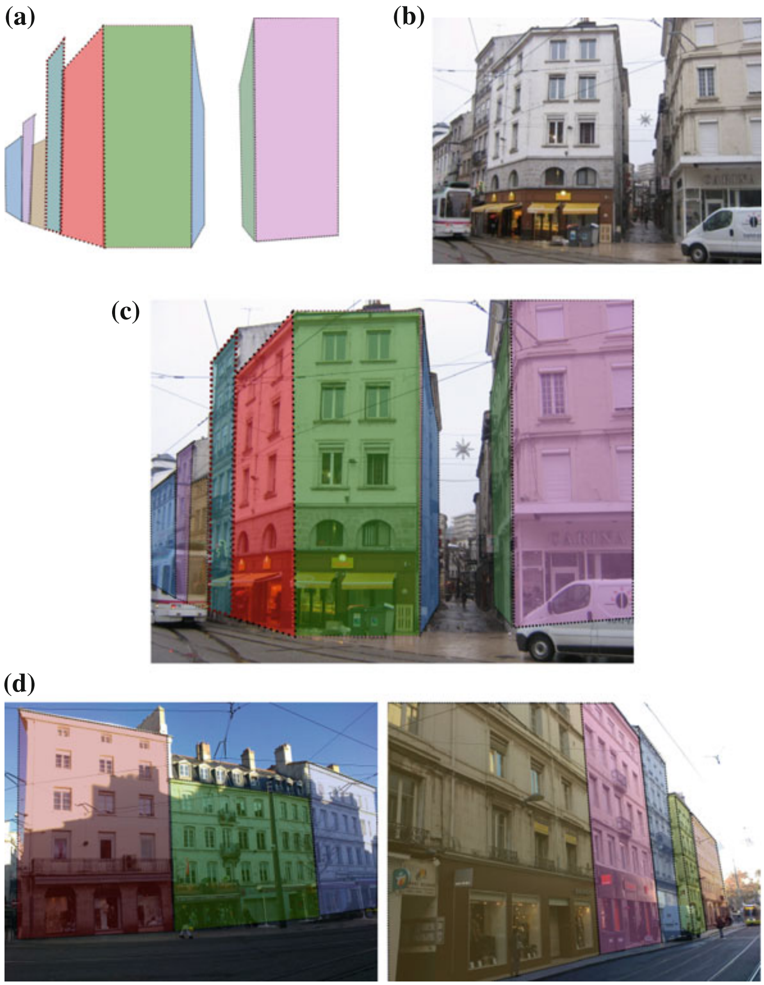


Fig. 14 ISOVIST result with real image. **a** isovist image from GIS data, **b** real image, **c** real image and isovist image coupling, **d** other examples

5.2 3D Space Syntax

Space Syntax denotes a group of indices like ‘visibility’, ‘openness’, ‘enclosure’ and ‘scale’ based on the calculation of isovists. Theses indices are proven to have a correlation with human perception (Putra and Yang 2005; Putra 2005. “Space Open Index” is introduced in (Fisher-Gewirtzman 2005), this index presents the volume of free spaces potentially seen from a given point. More visibility indices are presented in a 3D environment (Bilsen 2010). The author calculates statistics of the distance to

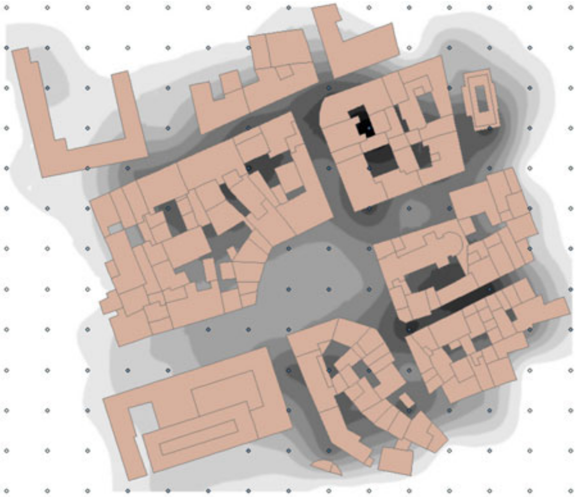


Fig. 15 Open space at 100 m, the clear area is high open sky visibility

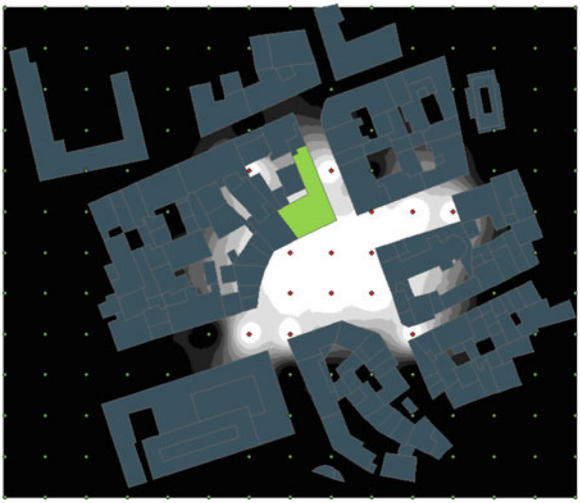


Fig. 16 Visibility area for the *green* building. The clear part is the visible region produced by the interpolation of points which can see the *green* building (the *red dots*)

3D environment in different horizontal plane (like minimum, standard of deviation), as well as the solid angular fraction of visible sky.

The method presented in this paper provides the opportunity to compute more Space Syntax indices in the 3D environment than the method presented in Bilsen (2010). Figure 15 presents the open space (the sky view) on a 100 m map. The dark part corresponds to the lower value for the open space solid angle.

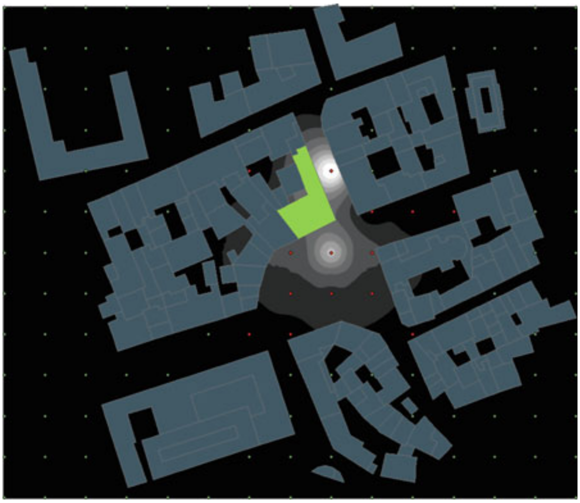


Fig. 17 Visibility area for the *green* building weighted with the visibility solid angle, the clearest part corresponds to the highest visibility solid angle and points which can see the *green* building (the *red dots*)

This calculus is called "Space Open Index" in Fisher-Gewirtzman (2005) who uses ray tracing technique.

Figure 16 presents the visible area for the green building.

Figure 17 presents the visible area for the green building, and it is weighted with the solid angle value for the visibility of the green building. This map presents the importance of the green building in the global view from a given point.

Actually the high value of the solid view angle means the object takes an important place in the visible space around the observer and as a result, the observer can see more details from this object. This represents a difference between two observers who can see the same object. If we suppose this building is an historical one with lovely facades a good view on this building from an apartment increase its value.

6 Conclusion

In this article we present a new algorithm for computing an Isovist in a 2D and 3D environment with a digital evaluation model (DEM). It permits the calculation of new Space Syntax indices, based on the content of the view like the percentage of a historic building that we can see from a vantage point or the percentage of the green area when we add a vegetation layer.

The method of ray tracing is faster than our method because it uses hardware technologies to calculate visible polygons in the 3D environment. Our method is a new way of visibility calculation. A real evaluation of this proposition needs a proper implementation in the hardware or in a low level programming language.

References

Ashihara Y (1984) The aesthetic townscape. MIT Press, Cambridge, pp 195–139

Batty M, Jiang B (1999) Multi-agent simulation: new approaches to exploring space-time dynamics in GIS. http://eprints.ucl.ac.uk/268/

Benedict ML (1979) To take hold of space: isovists and isovists fields. Environ Plann B 6:47–65

Bentley JL, Ottmann TA (1979) Algorithms for reporting and counting geometric intersections. IEEE Trans Comput C-28:643–647

Bilsen V (2009) How can serious games benefit from 3D visibility analysis? Presented at the international simulation and gaming association conference, Singapore

Bilsen V (2010) 3D visibility analysis in virtual learning environments and interactive and digital media. Presented at the interactive & digital media for education in virtual learning environment, New York

Boehm J, Haala N, Kapusy P (2002) Automated appearance-based building detection in terrestrial images. In: ISPRS commission v symposium, international archives on photogrammetry and remote sensing, vol 34, pp 491–495

Brossard T, Joly D, Tourneux F (2008) Modélisation opérationnelle du paysage. Paysage et information géographique, Lavoisier, pp 117–137

Christenson M (2010) Registering visual permeability in architecture: isovists and occlusion maps in AutoLISP. Environ Plann B Plann Des 37:1128–1136

Cipolla Drummond (2000) Vision algorithms: theory and practice. Springer, Berlin

Conroy Dalton R, Bafna S (2003) The syntactical image of the city: a reciprocal definition of spatial elements and spatial syntaxes. http://eprints.ucl.ac.uk/1104/

David P, Dementhon D, Duraiswami R, Samet H (2004) SoftPOSIT: simultaneous pose and correspondence determination. Int J Comput Vision 59(3):259–284

De Floriani L, Magillo P (1994) Abstract visibility algorithms on triangulated digital terrain models. Int J Geogr Inform Syst 8(1):13–41

Do EY-L (1994a) Design and description of form—using tool command language Tk/Tcl to visualize isovist by lighting and shadow casting analogy

Do EY-L (1994b) Isovist calculation in AutoCAD

Do EY-L (1995) Visual analysis through Isovist—building a computation tool

Do EY-L (1997) Tools for visual and spatial analysis of CAD models. CAAD futures 1997 conference, pp 373–388

Drummond T, Cipolla R (2002) Real-time visual tracking of complex structures. IEEE Trans Pattern Anal Machine Intell 24:932–946

Fisher-Gewirtzman D, Shach Pinsly D, Wagner IA, Burt M (2005) View-oriented three-dimensional visual analysis models for the urban environment. Urban Des Int 10:23–37

Fishman J, Haverkort H, Toma L (2009) Improved visibility computation on massive grid terrains. Presented at the (2009)

Floriani LD, Magillo P (2003) Algorithms for visibility computation on terrains: a survey. Environ Plann B Plann Des 30:709–728

Franklin WR, Ray CK (1994) Higher isn't necessarily better: visibility algorithms and experiments. In: Advances in GIS research: sixth international symposium on spatial data handling, vol 5, pp 751–770

Gibson JJ (1983) The senses considered as perceptual systems. Greenwood Press Reprint, Westport

Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press, Cambridge

Ittelson W (1960) Visual space perception, vol. 212. Springer, New York, pp. 6: Science 133:1241–1242 (1961)

Lake IR, Lovett AA, Bateman IJ, Day B (2000) Using GIS—and large-scale digital data to implement hedonic pricing studies. Int J Geog Inform Sci 14:521

Lynch K (1976) What time is this place? The MIT Press, Cambridge

Morello E, Ratti C (2009) A digital image of the city: 3D isovists in Lynch's urban analysis. Environ Plann B Plann Des 36:837–853

Ortegón-Aguilar J, Bayro-Corrochano E (2006) Lie algebra and system identification techniques for 3D rigid motion estimation and monocular tracking. J Math Imaging Vision 25:173–185

PUTRA SY (2005) GIS-based 3D volumetric visibility analysis and spatial and temporal perceptions of urban space

Putra SY, Yang PP-J (2005) Analysing mental geography of residential environment in Singapore using GIS-based 3D visibility, analysis

Pyysalo U, Oksanen J, Sarjakoski T (2009) Viewshed analysis and visualization of landscape voxel models. In: 24th international cartographic conference, Santiago, Chile

Rana S (2006) Isovist analyst: an arcview extension for planning visual surveillance. http://eprints.ucl.ac.uk/2104/

Rosenberg Y, Werman M (1998) Real-time object tracking from a moving video camera: a software approach on a PC. In: IEEE workshop on applications of computer vision, pp 238–239

Sourimant G, Morin L, Bouatouch K, De Rennes (2009) GPS, GIS and video registration for building reconstruction. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.2400

Suleiman W, Joliveau T, Favier E (2011) 3D urban visibility analysis with vector GIS data. Presented at the GISRUK 2011, University of Portsmouth, UK, pp 27–29, 26 April 2011

Turner A, Doxa M, O'Sullivan D, Penn A (2001) From isovists to visibility graphs: a methodology for the analysis of architectural space. Environ Plann B 28:103–121

Van Kreveld M (1996) Variations on sweep algorithms: efficient computation of extended viewsheds and class intervals. In: Proceedings of the 7th international symposium on spatial data handling, pp 13–15

Zhao C, Shi W, Deng Y (2005) A new Hausdorff distance for image matching. Pattern Recogn Lett 26:581–586

A Modeling Approach for the Extraction of Semantic Information from a Maritime Corpus

Dieudonné Tsatcha, Eric Saux and Christophe Claramunt

Abstract This paper introduces an algorithm for retrieving semantic information from a maritime corpus. The method is based on Natural Language Processing (NPL) and combines a segmentation of large documents with principles of a conceptual vector model (CVM) and synsets of words. This research is applied to the context of intelligent transport systems and maritime navigation. Based on documents regulating maritime traffic, this approach proposes an aid for navigational decision-making while significantly reducing the number of entities and relations required in the modeling process.

Keywords Natural language processing · Conceptual vector model · Semantics · Navigational decision aid

1 Introduction

Security for navigation in the maritime context is a significant challenge, which has been the focus of much research and developments even though much work remains to accomplish. Intelligent transport systems (ITS) provide some solutions (e.g., 3-dimensional GIS applied to maritime navigation[1] (Goralski and Gold 2008), Automatic Identification System (AIS)) but identification of a modeling approach which takes into account all the environmental components is not easy to achieve. This is mainly due to the fact that the maritime navigation space is complex (e.g., traffic regulation rules, restricted areas) and changes dynamically (tides, currents, winds) this having a direct incidence on navigation. Moreover, legibility (fog, day, night) can

[1] http://www.geovs.com/

D. Tsatcha (✉) · E. Saux · C. Claramunt
GIS group, Lanvéoc-Poulmic, Naval Academy Research Institute,
CC 600, F-29240 Brest Cedex 9, France
e-mail: dieudonne.tsatcha@ecole-navale.fr

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_12,

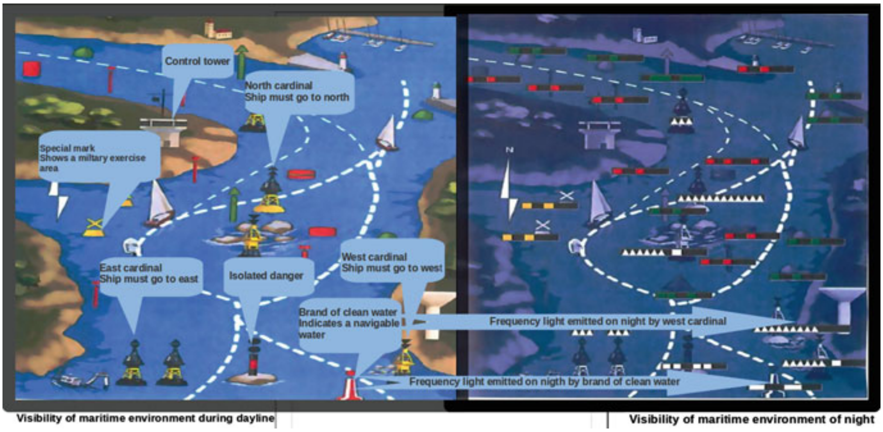

Control tower
North cardinal
Ship must go to north
Special mark
Shows a miltary exercise area
East cardinal
Ship must go to east
Isolated danger
West cardinal
Ship must go to west
Brand of clean water
Indicates a navigable water
N
Frequency light emitted on night by west cardinal
Frequency light emitted on nigth by brand of clean water
Visibility of maritime environment during dayline
Visibility of maritime environment of night

Fig. 1 An example of semantic information that influences the route of vessels by day (on the *left*) and by night (on the *right*). By day, the objects are recognized based on their shape and colors whereas they are identified by their light signal by night (Néméta 2008)

influence the perception of the actions to perform. Most ITS try to reduce collisions by improving the visual representation of the environment (Electronic Chart Display Information Systems, Automatic Radar Plotting Aids) (Claramunt et al. 2005) or analyzing information coming from sensors (Global Positioning System, Automatic Identification System, Radio Detection And Ranging (RADAR), infrared cameras, Lang Range Identification and Tracking (LRIT)). However and to the best of our knowledge, none of these methods facilitate significantly navigation planning. One objective is to reduce the complexity of representation by interpreting the semantic information generated by one object or several objects available in a maritime map and within a given area at a given time.

The main objective of this research is to build a decision platform for a maritime navigation environment based on some available semantics proposing a safe route to the captain. The assumption made is that the decision platform can significantly improve the captain's cognitive abilities during a high stress or high workload situation. We assume that a route preserves navigation safety if it takes into account the semantics and affordances of all the objects around the ship. An affordance is a quality of an object, or an environment, which allows an individual to perform an action (Gibson 1977). This implies to take into account restricted areas, the effect of ocean currents, the wind, radio signals and so on. Considering the knowledge that emerges from exterior events and the behavior of the objects located in the vicinity of the ship or those detected from sensors, a sailor should quickly get directional information regarding the route to follow and make the appropriate motion decision. One of the difficulties for the identification of entities is to take into account their salience. For example, a buoy may be expressed differently at night or during the day (cf. Fig. 1).

The development of a decision platform implies modeling a maritime environment and defining a relevant ontology. Maritime knowledge is extracted from documents used to regulate maritime navigation (U.S. Department of Transportation 2011; International Hydrographic Bureau 2011; Pearson 2008; James and Vicki 2011; National Oceanic and Atmospheric Administration 2011). Natural language processing is applied at two levels. The first level permits to extract the terms of the domain. The use of Yatea[2] software (Aubin and Hamon 2006) coupled with Treetagger (Schmid 1995) extracts the terms with the largest number of occurrences in the documents. The ones with a small number of occurrences, but which are important in navigation decisions, are also considered. The corpus contains 16,010 sentences defined with 413,076 words (175,578 nouns, 26,922 verbs, 20,703 adjectives, 9,106 adverbs). The second level extracts the semantics of the objects in the finite collection of states set by an expert. This extraction is possible thanks to conceptual vector applied to the sentences that relate to an object and projected in a decision space. By extracting the semantics of the objects, one can find the decision or the future area that the ship should follow. This paper mainly focuses on the second level assuming the first one to be a preprocessing step.

The remainder of the paper is organized as follows. Section 2 introduces the main principles of natural language processing for information retrieval in a general context before introducing the theoretical concepts of conceptual vectors applied to a word and a sentence. Section 3 develops our semantic extraction approach based on the definition of a decision space where sentences are projected within in order to associate the right semantics to a given concept. Section 4 presents a case study and applies our strategy for the extraction of semantics from concepts derived from maritime navigation documents. Finally, Sect. 5 draws some conclusions and outlines further work.

2 Theoretical Concepts of Conceptual Vector in Natural Language Processing

Information retrieval (hereafter IR) is an interdisciplinary research domain. Research in IR evolved over time and from early works in the 1960s with language indexation experiments (Cleverdon 1962). In the 1990s, retrieval engines were mainly based on the concept of keyword and without adequate representation of content for both documents and queries (Salton and MacGill 1983). Nowadays, recent progress consists in merging NPL (extracting the lexico-semantic structure of documents) and IR (indexing, matching, etc.) to find the semantic information related to a query (Strzalkowski et al. 1994; Dumais et al. 1997; Potthast et al. 2008). Information retrieval supports three basic processes (Hiemstra 2001): representation of the content of documents, representation of query and comparison of the two previous representations. In order to improve the information retrieval efficiency, documents are transformed into a

[2] Yatea is a free piece of software used for lexical disambiguisation of documents.

suitable representation. Becker et al. (2004) introduced the different representations that can be used and describes the relations between representations and models. The three most used models in IR research are the vector space model, the probabilistic model and the inference network model (Singhal 2001). Most systems assign a numeric score to every document and rank it using this score and do not take into account the semantic relatedness between query and sentences which satisfy the query. Tsatsaronis and Panagiotopoulou (2009) points out the importance of capturing semantics betweeen terms in IR. In this paper, we propose an algorithm developed from the concept of conceptual vector and disambiguisation where the relevance of results depends on this semantic relatedness.

The proposed algorithm is grounded on concepts of conceptual vector of word initially proposed by Lafourcade et al. (2002). Conceptual vectors have been mainly used for information retrieval and for meaning representation in the latent semantic indexing (LSI) model from latent semantic analysis (LSA) studies in psycholinguistics (Salton and MacGill 1983). Our approach is inspired from Chauché (1990), which proposes a formalism for the projection of the linguistic notion of semantic field in a vectorial space. A conceptual vector (or vector of concepts) of a word is a set of words in which each word determines a concept where this word can be employed. A conceptual vector of a sentence includes all concepts of the sentence. The latter is based on the direct sum of conceptual vectors of words composing the sentence.

2.1 Conceptual Vector of Word

The definition of a conceptual vector of a word is based on the concept of synset of a word. A synset of a word represents a concept and contains a set of interchangeable words, each of them having the same sense that names the concept (Beckwith et al. 1990). Another sense that names the concept defines another synset of the same initial word. Each word composing the synset that is different from the initial one is called *candidate word*. The definition of the conceptual vector associated to a word is based on a set of synsets and a metric measuring the distance between this word and each candidate word of a synset. The conceptual vector of a word is organized according to grammatical categories (adjective (a), adverb (r), noun (n), verb (v)) that the word may belong to and according to decreasing distance values inside a grammatical category. We use the distance defined in RiWordnet[3] (Daniel 2008), developed for creativity support in computation literature proposed by Daniel Howe.

More formally, let w be a word, $S^c = (s_i^c)_{i=1}^{n^c}$ the sets of n^c synsets of w and $C^c = (c_i^c)_{i=1}^{m^c}$ the sets of m^c candidate words of all synsets in category c then the conceptual vector $V(w)$ of the word w is defined as a weighted union of candidate words expressed in each grammatical category:

[3] RiWordnet is an API to WordNet that is a lexical database for the English language.

$$V(w) = \left(\bigcup_{l=1}^{m^a} c_l^a \delta(c_l^a, w)\right)_a \left(\bigcup_{l=1}^{m^r} c_l^r \delta(c_l^r, w)\right)_r \left(\bigcup_{l=1}^{m^n} c_l^n \delta(c_l^n, w)\right)_n \left(\bigcup_{l=1}^{m^v} c_l^v \delta(c_l^v, w)\right)_v \quad (1)$$

For illustration purposes, let us compute the conceptual vector of the word "port". Let $S^a = (s_i^a)_{i=1}^1$, $S^r = \emptyset$, $S^n = (s_i^n)_{i=1}^5$ and $S^v = (s_i^v)_{i=1}^8$ be the different sets of synsets of this word extracted from WordNet where (see Fig. 2):

- Adjective (a)
  - s_1^a: port, larboard (located on the left side of a ship or aircraft)
- Noun (n)
  - s_1^n: port (a place (seaport or airport) where people and merchandise can enter or leave a country)
  - s_2^n: port, port wine (sweet dark-red dessert wine originally from Portugal)
  - s_3^n: port, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)
  - s_4^n: larboard, port (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)
  - s_5^n: interface, port ((computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals))
- Verb (v)
  - s_1^v: port (put or turn on the left side, of a ship) "port the helm"
  - s_2^v: port (bring to port) "the captain ported the ship at night"
  - s_3^v: port (land at or reach a port) "The ship finally ported"
  - s_4^v: port (turn or go to the port or left side, of a ship) "The big ship was slowly porting"
  - s_5^v: port (carry, bear, convey, or bring) "The small canoe could be ported easily"
  - s_6^v: port (carry or hold with both hands diagonally across the body, especially of weapons) "port a rifle"
  - s_7^v: port (drink port) "We were porting all in the club after dinner"
  - s_8^v: port (modify (software) for use on a different machine or platform)

The sets of candidate words for word "port" are defined by C^a=(larboard), $C^r = \emptyset$, C^n=(embrasure,porthole,larboard,interface) and $C^v = \emptyset$. $C^r = \emptyset$ or $C^v = \emptyset$ means that there is no candidate word, i.e., there is no other sense than the initial one conveyed by the word "port". Finally, the conceptual vector of word "port" is given by:
V(port)=(larboard[1.00])$_a$(embrasure[1.00]porthole[1.00]larboard[1.00] interface[1.00])$_n$

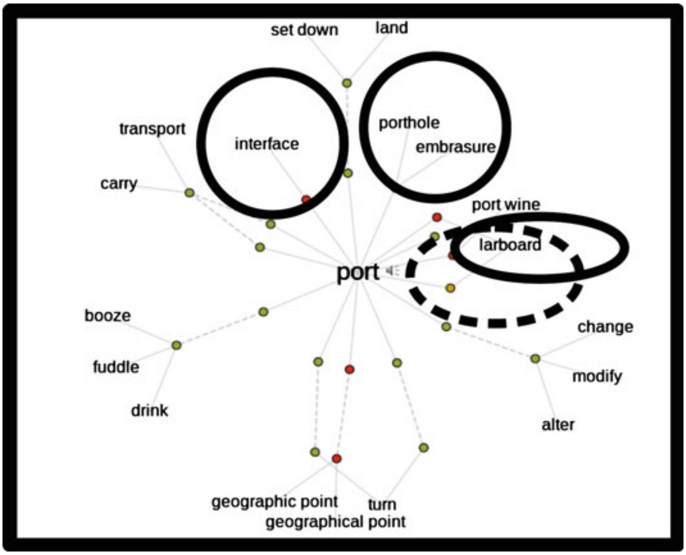


Fig. 2 Illustration of the synsets of the word "port" from a visual thesaurus (http://www.visualthesaurus.com/landing/): the circle with *continuous line* represents the noun synsets while the circle with *dashed line* represents the adjective synset

The distance δ between two words w_1,w_2, or between a word and a candidate word in the context of conceptual vectors, is computed as follow. Let G={adjective (a), adverb (r), noun (n), verb (v)} be the set of grammatical categories in the WordNet dictionary and P={P^a,P^r,P^n,P^v} a set of common parents of these words in the WordNet lexical network. w_1 and w_2 have a common parent if they share some semantic relations (hypernym, hyponym, holonym, troponym etc.). The distance between these two words is defined as[4]:

$$\delta(w1, w2)^6 = \begin{cases} 1 & \text{if } w_1 \text{ and } w_2 \text{ don't have a common parent} \\ \| \left[\frac{min(d(w_1, P^g), d(w_2, P^g))}{min(d(w_1, P^g), d(w_2, P^g)) + d(P^g, R)} \right]_{g \in G} \| \end{cases} \tag{2}$$

where $d(w_1, w_2)$ is the number of arcs between nodes w_1 and w_2, R the root node of the lexical network and $\|\|$ is the infinity norm. The shorter the distance between two words, the higher the semantic proximity between them.

Figure 3 represents an example of organisation of words in the WordNet dictionary for an arbitrary grammatical category $g = v$. In such a graph, the distance $\delta(word1, word2)$ between the two words is equal to $\| \frac{1}{1+1} \|$=0.5.

[4] http://www.rednoise.org/rita/wordnet/documentation/riwordnet_method_getdistance.htm

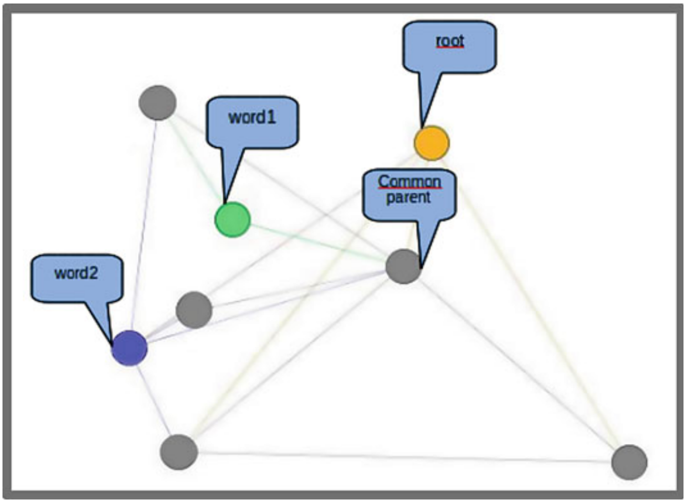


Fig. 3 Illustration of the structure of the lexical network used by WordNet

2.2 Conceptual Vector of Sentence

Assuming the principle that a sentence is a collection of polysemic words, we define the conceptual vector of a sentence as the direct sum of conceptual vectors of the words that the sentence contains. For each conceptual vector of a word, we take into account the grammatical category which is the same as the part-of-speech of this word in the sentence. In WordNet, the synsets of a word are computed for four grammatical categories (adjective, adverb, noun, verb). For a word where synset extraction is not possible from its part-of-speech, all the conceptual vectors are determined regardless of its part-of-speech in the sentence. A conceptual vector of a sentence gathers all concepts that the sentence relates. In the definition the conceptual vector does not take into account the type (i.e., declarative, exclamatory, interrogative, imperative) and form (i.e., affirmative/negative, active/passive, neutral/emphatic) of a sentence. Let s be a sentence composed of n words w_i, then the conceptual vector of the sentence s is defined by:

$$V(s) = \sum_{i=1}^{n-1} V(w_i) \oplus V(w_{i+1}) \tag{3}$$

The direct sum between two conceptual vectors of a word is defined as a union of candidate words organized by grammatical category, each one weighed by its distance to the word that it represents (cf. Eq. 2). The direct sum of conceptual vectors is usually used to extend the field of concepts of the working space (i.e., the context).

Table 1 Disambiguisation of the sentence

Word	Part-of-speech[a]	Lemma
The	DT	the
ship	NN	ship
is	VBZ	be
in	IN	in
the	DT	the
port	NN	port

[a] http://en.wikipedia.org/wiki/Brown_Corpus#Partofspeech_tags_used

Let w_i and w_j be two words and $V(w_i)$ and $V(w_j)$ their corresponding conceptual vectors, the direct sum between these two conceptual vectors is expressed by the equation:

$$\begin{aligned} &V(w_i) \oplus V(w_j) \\ &= \left(\bigcup_{k=i,j}\left(\bigcup_{l=1}^{m^a_{w_k}} c^a_{l,w_k}\delta(c^a_{l,w_k}, w_k)\right)\right)_a \left(\bigcup_{k=i,j}\left(\bigcup_{l=1}^{m^r_{w_k}} c^r_{l,w_k}\delta(c^r_{l,w_k}, w_k)\right)\right)_r \\ &\quad \left(\bigcup_{k=i,j}\left(\bigcup_{l=1}^{m^n_{w_k}} c^n_{l,w_k}\delta(c^n_{l,w_k}, w_k)\right)\right)_n \left(\bigcup_{k=i,j}\left(\bigcup_{l=1}^{m^v_{w_k}} c^v_{l,w_k}\delta(c^v_{l,w_k}, w_k)\right)\right)_v \end{aligned} \tag{4}$$

In the particular case where $w_i = w_j$, $V(w_i) \oplus V(w_i) = V(w_i)$.

As an example, let us consider the following sentence s="The ship is in the port". The initial disambiguisation of this sentence is proposed in Table 1:

The conceptual vectors of the different words in a sentence s according to their part-of-speech are:

- V(the)=∅
- V(ship)=∅
- V(is)=(be[0.00]exist[0.00]equal[0.00]constitute[0.00]represent[0.00] comprise[0.00] follow[0.00]embody[0.00]personify[0.00]live[0.00]cost[0.00])$_v$
- V(in)=(inwards[1.00]inward[1.00])$_n$(inch[0.00]indium[0.00])$_r$
- V(the)=∅
- V(port)=(embrasure[1.00]porthole[1.00]larboard[1.00]interface[1.00])$_n$

The direct sum between V(ship) and V(port) is:

V(ship) $\oplus V$(port) = (embrasure[1.00]porthole[1.00]larboard[1.00]interface[1.00])$_n$

The resulting normalised (see Sect. 3.1) conceptual vector is:

$$\begin{aligned}
V(\text{“The ship is the in the port”}) &= V(\text{the}) \oplus V(\text{ship}) \oplus V(\text{is}) \oplus V(\text{in}) \oplus V(\text{the}) \\
&\quad \oplus V(\text{port}) \\
&= (\text{inch}[0.00]\text{indium}[0.00])_r \\
&\quad (\text{inwards}[0.38]\text{inward}[0.40]\text{embrasure}[0.47] \\
&\quad \text{porthole}[0.52]\text{larboard}[0.58]\text{interface}[0.66])_n \\
&\quad (\text{be}[0.00]\text{exist}[0.00]\text{equal}[0.00] \\
&\quad \text{constitute}[0.00]\text{represent}[0.00]\text{comprise}[0.00] \\
&\quad \text{follow}[0.00] \\
&\quad \text{embody}[0.00]\text{personify}[0.00]\text{live}[0.00]_v \\
&\quad \text{cost}[0.00])
\end{aligned}$$

3 Decision Space for the Extraction of Semantic Information

The goal of this section is to extract the semantic information related to a concept. We introduce a decision space where the different conceptual vectors of the sentences which describe this concept must be projected. The decision space contains a list of feasible options identified in the macro-phases of the decision strategy. Jankowski and Nyerges summarized the macro-phase of the decision strategy in three steps (Jankowski and Nyerges 2003): (1) intelligent about the values, objectives and criteria (2) design of a set of feasible options, (3) choice about recommendations. The feasible options should be linked to the objectives and validated by an expert. The conceptual vectors of words that are correlated to the semantics we are searching for define the semantic axes (i.e., the basis) of the decision space.

3.1 Projection of a Sentence in a Decision Space

The projection of a sentence s in a decision space corresponds to the projection of the candidate words of the conceptual vector of s (i.e., $V(s)$) in order to valuate the contribution in each semantic direction of the basis. We thus derive the principal direction detailed in the next subsection.

Let us assume that one want to compute the contribution x_i^c of a candidate word of $V(s)$ in the semantic direction d and category c where n is the number of candidate words of the conceptual vector $V(d)$ that defines a semantic axis of the decision space. If c_i^c is the common candidate word of the two conceptual vectors $V(s)$ and $V(d)$ with weights δ_s and δ_d respectively then

$$x_i^c = \frac{(1 - \delta_s * \delta_d)}{n}. \tag{5}$$

x_i^c equals to zero if the candidate word of $V(s)$ does not belong to $V(d)$. In Eq. 5, the value is weighed by the number of candidate words in the semantic axis considering that a candidate word of the conceptual vector of a sentence $V(s)$ has a higher influence if the number of candidate words of $V(d)$ is low. A candidate word of $V(s)$ having a weight equal to 1 (i.e., a poor semantic contribution) may have no contribution in a semantic axis (i.e., $x_i^c = 0$ if $\delta_s = 1$ and $\delta_d = 1$) and is not taken into account in the final decision. To tackle this problem, one can normalise the conceptual vector of a sentence (Sect. 2.2), discarding the case where a weight is equal to 1.

The contribution of a sentence in a category c is the sum of the contribution of the m_c candidate words of this category, i.e., $x^c = \sum_{i=1}^{m_c} x_i^c$. Finally, the contribution of a sentence is the sum of the contributions in each category, i.e., $x = x^a + x^r + x^n + x^v$. The higher the semantic contribution in a direction, the higher the projection value x. This process is repeated in all the semantic directions that contribute to the decision space.

Let us illustrate this principle with the following example where one want to find the contribution of the word "stay" in the semantic direction "stop". We firstly define the conceptual vectors of these two words:

$$\begin{aligned}
V(\text{stop}) = {} & (\text{halt[0.00]block[0.00]check[0.00]} \\
& \text{arrest[0.00]blockade[0.12] bar[0.14] end[0.14] finish[0.14]} \\
& \text{barricade[0.22]} \\
& \text{break[0.29]cease[0.67]intercept[0.73]kibosh[1.00]} \\
& \text{terminate[1.00]contain[1.00]quit[1.00]discontinue[1.00]})_v \\
& (\text{halt[0.00]stoppage[0.00]stopover[0.00]} \\
& \text{layover[0.00]arrest[0.00]check[0.00]hitch[0.00]stay[0.00]} \\
& \text{occlusive[0.00]plosive[0.00]period[0.00]point[0.00]} \\
& \text{diaphragm[0.00]catch[0.00]blockage[0.00]block[0.00]} \\
& \text{closure[0.00]occlusion[0.00]})_n \\
V(\text{stay}) = {} & (\text{remain[0.00]rest[0.00]stick[0.00]} \\
& \text{bide[0.00]abide[0.00]continue[0.00]detain[0.00]delay[0.00]} \\
& \text{persist[0.00]outride[0.00]quell[0.00]appease[]})_v \\
& (\text{arrest[1.00]check[0.33]halt[0.33]stop[0.33]hitch[0.50]} \\
& \text{stoppage[1.00]})_n
\end{aligned}$$

Secondly, we compute the projection values of the candidate words of $V(\text{stay})$ in each category. Projection is always null except for common candidate words c_1^n=arrest, c_2^n=check, c_3^n=halt, c_4^n=hitch, c_5^n=stoppage. Projections values are computed as follow:

$x_1^n = \frac{1-0.00*1}{35} = 0.029$, with c_1^n = arrest and $n = 35$, $\delta_{stop} = 0.00$, $\delta_{stay} = 1$

$x_2^n = \frac{1-0.00*0.33}{35} = 0.029$, with c_2^n = check and $n = 35$, $\delta_{stop} = 0.00$, $\delta_{stay} = 0.33$

Table 2 Projection values of sentences that refer to the concept "low water" in the decision space (V(maneuver), V(stop), V(continue), V(back))

$\frac{Affordance}{Sentence}$	Maneuver	Stop	Continue	Back
Total	0.12	**0.12**	0.02	0.12

$$x_3^n = \frac{1-0.00*0.33}{35} = 0.029, \text{ with } c_3^n = \text{halt and } n = 35, \delta_{stop} = 0.00, \delta_{stay} = 0.33$$

$$x_4^n = \frac{1-0.00*0.50}{35} = 0.029, \text{ with } c_4^n = \text{hitch and } n = 35, \delta_{stop} = 0.00, \delta_{stay} = 0.50$$

$$x_5^n = \frac{1-0.00*1}{35} = 0.029, \text{ with } c_5^n = \text{stoppage and } n = 35, \delta_{stop} = 0.00, \delta_{stay} = 1$$

For category noun, we deduce that the contribution of word "stay" in the direction "stop" is: $x^n = x_1^n + x_2^n + x_3^n + x_4^n + x_5^n = 0.15$. It results that the final contribution value is: $x = x^a + x^r + x^n + x^v = 0.15$.

3.2 Principal Semantic Direction for a Concept

This section aims at determining the principal semantics in a decision space that is associated to a concept. The first stage of the process consists in identifying the sentences of the corpus related to this concept and to project each of them in the decision space. The next stage focuses on the computation of the main contribution of these sentences. The contribution of the sentences in one semantic axis of the decision space is the sum of the contributions of the sentences regarding this direction. We apply this principle in all the semantic directions that contribute to the decision space. The semantic direction which warrants the highest trust is the one which has the highest coordinate or score. This semantic direction is called the principal semantic direction associated to the concept.

In the case where at least two directions have the same score, the direction which ensures security of mariner is considered. Regarding experts, these decisions are classified according to the decreasing security order: back, stop, maneuver and continue. As a result in Table 2, the decision corresponding to the "low water" concept is "to go back".

4 Case Study

Let us illustrate our approach by the analysis of the semantics associated to the concept of "anchorage area". The sentences related to this concept in our corpus are:

s_1: "Any vessel anchored outside of the prescribed anchorage limits must move to a prescribed **anchorage area** when space becomes available".

Table 3 Projection values of sentences that refer to the concept “anchorage area” in the decision space $(V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back}))$

$\frac{Affordance}{Sentence}$	Maneuver	Stop	Continue	Back
s_1	0.00	0.01	0.00	0.00
s_2	0.01	0.00	0.00	0.01
s_3	0.01	0.00	0.00	0.01
s_4	0.00	0.01	0.00	0.00
s_5	0.01	0.04	0.01	0.00
Total	0.03	**0.06**	0.01	0.02

s_2: “Whenever, in the opinion of the captain of the port such action may be necessary, he may require any or all vessels in any designated **anchorage area** to moor with two or more anchors”.

s_3: “Reserved anchors shall be placed well within the anchorage areas, so that no portion of the hull or rigging will at any time extend outside of the **anchorage area**”.

s_4: “Except in cases where unforeseen circumstances create conditions of imminent peril, or with the permission of the captain of the port, no vessel shall be anchored in baltimore harbor and patapsco river outside of the **anchorage areas** established in this section for more than 24 h”.

s_5: “Any vessel anchoring, under great emergency, within this area shall be placed as close to an **anchorage areas** as practicable, and shall move away immediately after the emergency ceases”.

In a second stage, one decide to extract the behaviour that a mariner can decide facing to the concept “anchorage area”. We restrict our case study to the actions or affordances (continue, stop, go back and maneuver) that identify the four semantic axes of our decision space defined by the basis of conceptual vectors $(V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back}))$. As regards experts in maritime navigation, these actions describe the different actions a mariner can take in front of an object in the real word. Each of these actions or situations has the following meaning: maneuver indicates to the mariner that he must change his trajectory; stop indicates to the mariner that he must temporarily remain in his navigation area (for example, the vessel enters in an anchorage area or he receives a special signal which requires him to stop the navigation); back denotes that he must turn around because the environment becomes dangerous or impracticable (for example, in the presence of dense fog or strong storm); and continue proposes to mariner he can follow the same trajectory because none unsafe event is detected. The principal affordance we can associate to the concept “anchorage area” is *stop*, because it has the highest coordinate with value $0.06 = 0.01 + 0.00 + 0.00 + 0.01 + 0.04$. The coordinates of the projections of each sentence in the decision space are summarized in Table 3 and show a visualisation of this decision space in Fig. 4.

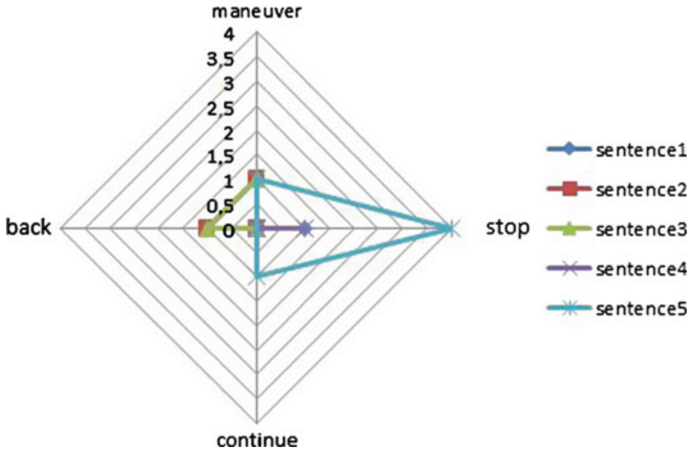


Fig. 4 Visualisation of the decision space for the concept "anchorage area"

Table 4 Projection values of sentences that refer to the concept "cardinal buoy" in the decision space $(V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back}))$

$\frac{Affordance}{Sentence}$	Maneuver	Stop	Continue	Back
s_1	0.02	0.00	0.00	0.00
s_2	0.02	0.01	0.00	0.00
Total	**0.04**	0.01	0.00	0.01

Let us illustrate our strategy with a second example in which we try to extract the affordance related to the concept "cardinal buoy". Accordingly and using the same documents previously cited, one selects the sentences:

s_1: "For example, a particular **cardinal buoy** represented through a symbol on a chart".

s_2: "The top marks of **cardinal buoys** consist of the combination of two black cones mounted one above the other on the top of the buoy with the following, combinations:

(a) both cones pointing up = North cardinal,
(b) both pointing down = South cardinal,
(c) one pointing up and the other down with their bases together = East cardinal,
(d) one pointing up and the other pointing down with their points together = West cardinal.

The principal affordance that can be associated to the concept "cardinal buoy" is *maneuver*, because it has the highest coordinate with value $0.04 = 0.02 + 0.02$. We summarize the coordinates of the projections of each sentence in the decision space in Table 4 and show a visualisation of this decision space in Fig. 5.

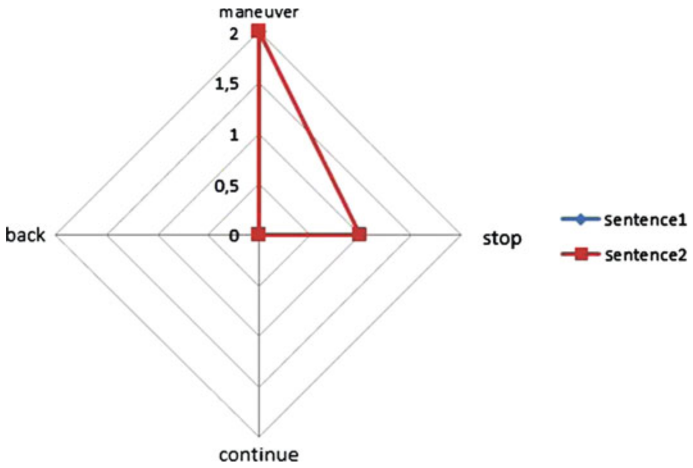


Fig. 5 Visualisation of the decision space for the concept “cardinal buoy”

4.1 Glosses of the Concepts of a Sentence

In some cases, the projection of a word or a sentence in a decision space generates a conceptual vector whose euclidean norm equals zero (i.e., each semantic contribution is equal to zero) and no decision emerges. We use the concept of gloss to improve the results. The glosses of a word are the different definitions of it. For example, a gloss of word “port” may be “a place (seaport or airport) where people and merchandise can enter or leave a country” (definition from synset s_1^n of Sect. 2.1). As a result when no decision is proposed, the principle of the semantic extraction strategy is to use the definition of a word (i.e., its gloss) to extract the semantic information related to it.

The new coordinate of a sentence whose conceptual vector is null is computed by using the glosses of the each word in this sentence. For each word w, we extract the different glosses and project them in our decision space (see Sect. 3.1). The most relevant gloss of word w is the gloss having the highest coordinate in the decision space. The infinity norm ($\|\|_\infty$) is applied to find the most contributing gloss of a word. Lastly the coordinates generated by each word of the initial sentence is sumed to get a new coordinate for it.

For example, two sentences in the corpus are related to the concept of “dense fog”:

s_1: “From April to September there are only a few days with **dense fogs**”.
s_2: “**Dense fog** is more common offshore and should be expected on unusually warm, humid winter and spring days”.

This implies that no decision is taken since the projection of the concept “dense fog” is null in each semantic axis, i.e.:

s_1: maneuver[0.00]stop[0.00]continue[0.00]back[0.00]
s_2: maneuver[0.00]stop[0.00]continue[0.00]back[0.00]

Table 5 Projection values of sentences that refer to the concept "dense fog" in the decision space $(V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back}))$ using the concept of gloss

$\frac{Affordance}{Sentence}$	Maneuver	Stop	Continue	Back
s_1	0.00	0.01	0.01	0.12
s_2	0.00	0.05	0.02	0.35
Total	0.00	0.06	0.03	**0.47**

Consequently, the glosses of the terms of sentences s_1 and s_2 are used in order to try to find a more accurate decision. The projections of the different glosses in the decision space give the results presented in Table 5 and lead to the decision to *go back*:

5 Conclusion and Further Work

This paper introduces a general strategy to extract semantic information from a corpus. We assume that the analysis of documents written by experts in a specific domain gives richer information than the exploitation of usual definitions found in common dictionaries. Accordingly, and in order to propose a vector of concepts of a word or a sentence, we use WordNet a lexical database of English where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Each component of this conceptual vector called "candidate word" is associated to a value which quantifies the semantic distance between the word or the sentence and its candidate word.

The extracted semantic information identifies the root of the second stage devoted to decision aid. The goal of this second stage is to extract the right decision with respect to a concept. The process applied is to define a decision space made up of different semantic axes in correlation with the application domain in which sentences are projected. The final decision is derived from the analysis of the main contribution observed in the semantic directions. This emphasizes the fact that the semantic richness of the initial corpus is important and influences the success of the strategy more than the choice of the WordNet dictionnary. To improve the results, the initial strategy is extended by considering not only the synsets of a word but also its glosses. The proposed strategy is applied to the extraction of semantic information in the maritime context for navigation aids but the process can easily be applied to other domains.

Further work concerns the development of a real time navigation aid platform which takes into account semantic information generated by objects (lighted buoy, water wayroute, radio signal, ships, etc.) or exterior events (wind, fog, stream, etc.) which appear in the vicinity of the ship. We assume that the descriptions and rules about these objects appear in the initial corpus used for disambiguisation.

This platform will be coupled with a spatio-temporal ontology of the maritime environment that will store the initial and the extracted knowledge.

Acknowledgments The authors grateful to the "Établissement Principal du Service Hydrographique et Océanographique de la Marine" (Brest-France) for their nautical documents and for their assistance in this study. Also thank to Thierry Hamon for his supporting in the use of software Yatea. We would also like to thank Mathieu Lafourcade for his introduction to conceptual vectors and Daniel Howe for his help on RiWordnet, and Alicino Ferreira for his careful proofreading of this article.

References

Aubin S, Hamon T (2006) Improving term extraction with terminological resources. In: Salakoski T, Ginter F, Pyysalo S, Pahikkala T (eds) Proceeding of the 5th international conference on NLP, FinTAL 2006, advances in natural language processing, pp 380–387. No. 4139 in LNAI, Springer, Aug 2006.

Becker J, Grob L, Hellingrath B, Klein S, Kuchen H, Müller-Funk U, Vossen G (2004) Advances in information systems and, management science, Logos Verlag Berlin GmbH, ISSN: 1611Ã¢â?¬â??3101.

Beckwith R, Fellbaum C, Gross D, Miller K (1990) Introduction to WordNet : an online lexical database. Int J Lexicogr 3(4):235–244

Chauché J (1990) Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. TAL Inf 31(1):17–24

Claramunt C, Fournier S, Li X, Peytchev E (2005) Real-time geographical information for ITS. In: Proceedings of the 5th IEEE international conference in intelligent transportation systems, pp 237–242.

Cleverdon CW (1962) Report on testing and analysis of an investigation into the comparatie efficiency of indexing systems.

Daniel CH (2008) A WordNet library for java processing. http://www.rednoise.org/rita/wordnet/documentation/index.htm

Dumais ST, Letsche TA, Littman ML, Landauer TK (1997) Automatic cross-language retrieval using latent semantic indexing. In: AAAI-97 spring symposium series: cross-language text and speech retrieval, pp 18–24. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.5717

Gibson JJ (1977) The theory of affordances. Lawrence Erlbaum, Hillsdale

Goralski R, Gold C (2008) Marine GIS: progress in 3D visualization for dynamic GIS. In: spatial data handling, Springer, pp 401–416.

Hiemstra D (2001) Using language models for information retrieval. Ph.D. thesis, Taaluitgeverij Neslia Paniculata, Jan 2001.

International Hydrographic Bureau (2011) MONACO: recommended ENC validation checks.

James M, Vicki G (2011) The handbook of delaware boating laws and responsabilities, By Boat Ed, a division of Kalkomey Enterprises. Inc , Texas

Jankowski P, Nyerges T (2003) Geographic information systems for group decision making: towards a participatory, geographic information science. Taylor & Francis, London

Lafourcade M, Prince V, Schwab D (2002) Vecteurs conceptuels et structuration émergente de terminologie. Traitement Automatiques des Langues 43(1):43–72

National Oceanic and Atmospheric Administration (2011) US department of commerce: United States Coast Pilot, 44th edn. (2011).

Néméta A (2008) Code Vagnon Permis Plaisance : Option cotière, Vagnon edn. (2008).

Pearson M (2008) Mémento Vagnon du Skipper : Moteur et voile (2008).

Potthast M, Stein B, Anderka M (2008) A Wikipedia-based multilingual retrieval model. In: Macdonald C, Ounis I, Plachouras V, Ruthven I, White RW (eds) Proceedings of the 30th European conference on IR research, ECIR 2008, advances in information retrieval, LNCS, vol 4956. Springer, Berlin, pp 522–530. http://dx.doi.org/10.1007/978-3-540-78646-7_51

Salton G, MacGill M (1983) Introduction to modern information retrieval. McGrawHill, New York

Schmid H (1995) Treetagger–a language independent part-of-speech tagger. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

Singhal A (2001) Modern information retrieval: a brief overview. IEEE Data Eng, Bulletin 24(4): 35-43.

Strzalkowski T, Carballo J, Marinescu M (1994) Natural language information retrieval: Trec-3-report. In: Proceedings of the 3rd text retrieval conference (1994).

Tsatsaronis G, Panagiotopoulou V (2009) A generalized vector space model for text retrieval based on semantic relatedness. In: Proceedings of the 12th conference of the European chapter of the association for, computational linguistics (EACL-09) April 2009.

U.S. Department of Transportation (2011) United States Coast Guard: Navigation Rules International-InLand.

Topological Adjustment of Polygonal Data

Jan Oliver Wallgrün

Abstract Qualitative spatial relations, in particular (mereo)topological relations such as those defined in the 9-Intersection model or RCC-8 and RCC-5, play an important role as constraints in many applications of spatial data processing such as conflation and data cleaning, map generalization, and solving layout problems in general. A fundamental problem in such applications is to adjust geometric data such that certain topological constraints are satisfied, while minimizing the changes that need to be made to the geometric input data. We develop an approach to solve this problem for topological relations between polygonal objects and with displacement of the objects being the only allowed adjustment operation. Our approach is based on a formalization of these relations as sets of (in)equations which can then be translated into a MNLP program and be solved using a dedicated MNLP solver. Our suggested adjustment algorithm uses Minkowski sums of the involved polygons to achieve a linear number of (in)equations per relation.

Keywords Topological relations · Integrity constraints · Adjustment theory · Qualitative spatial reasoning · Displacement · Data cleaning

1 Introduction

Many models for formalizing spatial relations and performing reasoning with them have been suggested and investigated in the literature. The resulting formalisms are often referred to as qualitative spatial calculi and investigated as part of a research area called Qualitative Spatial Reasoning (QSR) (Cohn and Hazarika 2001; Renz and Nebel 2007). From the different aspects of space formalized in these models,

J. O. Wallgrün (✉)
Department of Geography, GeoVISTA Center, The Pennsylvania State University, Pennsylvania, PA, USA
e-mail: wallgrun@psu.edu

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_13,

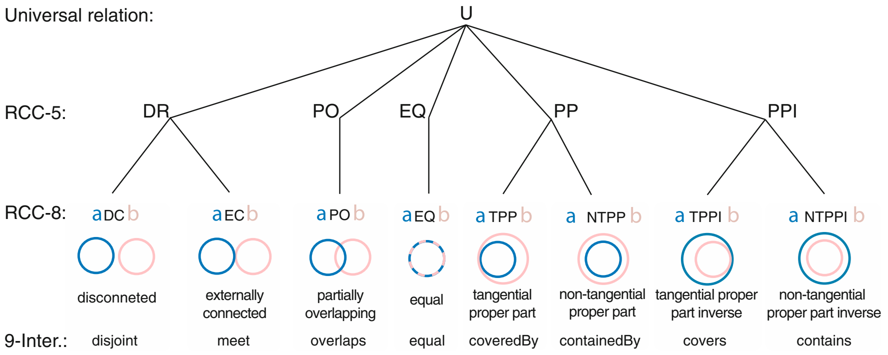


Fig. 1 Hierarchy of topological relations with the eight base relations distinguished by the of 9-Intersection and RCC-8 models at the *bottom* and the relations of the coarser RCC-5 model in the *middle*. At the *top* the all subsuming universal relation *U*

topological and mereotopological approaches such as the 9-Intersection model by Egenhofer (1991) and the RCC-8 model by Randell et al. (1992) have been of particular influence in different domains of spatial data handling. For simple regional objects in the plane, both formalisms define the eight binary topological relations shown at the bottom level of Fig. 1, for instance distinguishing whether two objects are complete disconnected (DC), externally connected (EC) meaning their boundaries touch, partially overlapping (PO), or whether one object is a tangential or non-tangential proper part of the other object (relations TPP and NTPP). Coarser models have also been proposed such as RCC-5 which groups together EC and DC, TPP and NTPP, and the inverses NTPPI and TPPI and, hence, only distinguishes five basic relations (middle level of Fig. 1).

The basic relations of the 9-Intersection and RCC models have, for instance, been utilized to describe spatial relationships in query and retrieval scenarios (Clementini et al. 1994), to formalize (geo)spatial concepts and processes (Klippel et al. 2008), and to specify background knowledge and integrity constraints in the context of spatial and spatio-temporal database applications (Haarslev and Möller 1997). When the relations of a qualitative spatial calculus are used to describe spatial integrity constraints—for instance that certain kind of objects may not overlap or that one object needs to be completely contained within a different object—it is often desirable to not only be able to detect when such constraints are violated in a given geometric data set but also be able to resolve such inconsistencies automatically, modifying the original data as little as possible. Such optimization problems of adjusting geometric data to a given set of not necessarily qualitative constraints arise in many application areas such as integration and conflation, data cleaning, and map generalization. While there exists ample work on realizing certain topological constraints, in particular non-overlap (Marriott et al. 2001), the general problem of adjusting geometric data to the relations from a topological or other qualitative calculus has not been studied in detail.

This work is a step in this direction with the goal of bridging between existing work on QSR and work on adapting geometric data to spatial constraints employing the general framework of constraint optimization and adjustment theory. We first define the general *qualitative adjustment problem* for arbitrary spatial calculi in Sect. 3. In the following, we focus on the mentioned topological calculi 9-Intersection and RCC, and the specific instance in which the geometric data consists of polygons in 2D space which may be translated/displaced but not change their shape and size. We develop an approach for formalizing the full set of topological relations defined in these calculi in terms of sets of (in)equations in Sect. 4. We utilize Minkowski sums between the involved polygons to achieve a formalization in which the number of (in)equations per relation grows only linearly wrt. the number of vertices. As described in Sect. 5, the resulting formalization of a problem instance is then translated into a mixed-integer non-linear programming (MNLP) problem and solved by a dedicated MNLP solver. Section 6 provides an example which demonstrates that our approach is indeed able to solve practical problems from the area of spatial data cleaning.

2 Related Work

The problem of adapting spatial data to meet various kinds of spatial constraints has received significant attention in several application domains of spatial data processing such as data integration and cleaning and map generalization. One commonly distinguishes different classes of spatial constraints such as graphical/metrical, topological, structural, etc. (see for instance Steiniger and Weibel (2007)). The rich spectrum of employed approaches ranges from local search (Ware and Jones 1998), over global optimization and least square adjustment techniques (Sester 2000; Harrie 1999), to agent-based frameworks (Lamy et al. 1999).

The approach closest to ours is the work described in Marriott et al. (2001). It also employs Minkowski polygons to formalize non-overlap constraints. Our approach extends this line of research by introducing two different kinds of Minkowski polygons which allows for describing all topological relations defined in the previously mentioned qualitative calculi. A formalization of RCC-8 relations in terms of systems of (in)equations has already been proposed in Bhatt et al. (2011). However, our Minkowski based approach has the advantage that it is much more efficient in terms of the required number of (in)equations per relation (linear versus quadratic).

3 The Qualitative Adjustment Problem

A qualitative spatial calculus defines a set $\mathcal{B}$ of *base relations* (e.g., DC, EC) over a domain D of spatial objects (e.g., points or regions in 2D) together with a set of operations that enable elementary reasoning. Spatial knowledge can then be expressed in the form of relational statements such as $A\{\mathrm{EC}\}B$. Incomplete

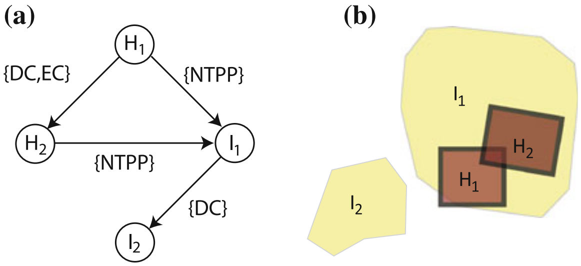


Fig. 2 Example of a qualitative adjustment problem: given (**a**) a qualitative constraint network over a spatial calculus $\mathcal{C}$ (here RCC-8) and (**b**) geometric data for the involved objects, adjust the geometric data until all relations from the network are satisfied while changing the geometries as little as possible (as specified by a cost function)

knowledge can be expressed using disjunctions of basic relations (here written as sets, e.g., $A\{\mathrm{DC}, \mathrm{EC}\}B$). If nothing is known about the relation between two objects the implicit relations is U standing for the universal relations which is the disjunction of all base relations.

A qualitative knowledge base can be illustrated as a so-called *qualitative constraint network* (QCN) (see Fig. 2a) with nodes standing for the objects and edges labeled by the respective qualitative relations (no connecting edge means the relation is U). The spatial relations of a qualitative calculus can be seen as constraints which restrict the possible geometries that the related objects may adopt. This makes them well suited to formulate integrity constraints. When applying a set of integrity constraints using qualitative relations from a calculus $\mathcal{C}$ to a given set of objects, the result is a QCN N over $\mathcal{C}$ that states which relations may hold between each pair of objects. The goal then is to adapt the geometric data such that it satisfies the relations in N, while minimizing a cost function over the changes made. We refer to this problem of adjusting geometric data to meet qualitative spatial integrity constraints as the *qualitative adjustment problem* (cmp. Fig. 2) and define it formally as:

Definition 1 (*qualitative adjustment problem*) Given (I) a set of geometric objects $\mathcal{O} = (O_1, ...O_n) \in D^n$ from a spatial domain D, (II) a QCN N over a calculus $\mathcal{C}$ with D as domain which has a node for each O_i of $\mathcal{O}$, and (III) a cost function $c : D \times D \rightarrow \mathbb{R}^+$ describing the costs for transforming object $o \in D$ into a new geometric object $o' \in D$, the *qualitative adjustment problem* is to compute a new set of geometric objects $\mathcal{O}' = (O'_1, ...O'_n) \in D^n$ that

$$(I) \quad \textit{minimizes the overall costs}: \quad \forall \mathcal{O}'' \in D^n : \sum_{i=1}^{n} c(O_i, O''_i) \geq \sum_{i=1}^{n} c(O_i, O'_i)$$

$$(II) \quad \textit{satisfies all constraints in } N: \quad \forall 1 \leq i, j \leq n : rel_{\mathcal{C}}(O'_i, O'_j) \subseteq C_{ij}$$

where $rel_{\mathcal{C}}(o, p)$ stands for the base relation from $\mathcal{C}$ holding between two objects $o, p \in D$ and C_{ij} is the constraint between O_i and O_j in N.

As apparent by the presence of the cost function c, the qualitative adjustment problem falls into the general category of constrained optimization problems. While we here restrict ourselves to relations from a single qualitative calculus, the definition can easily be generalized to sets of QCNs, each for a different calculus. In the remainder of this text, we focus on the topological constraints from Fig. 1 holding between simple polygonal objects in 2D and will use the relation names as defined in the RCC-8 and RCC-5 models. Moreover, we will develop a specific solution for the case in which geometric objects are only allowed to undergo specific transformations. In our case, only a translation/displacement is allowed, preserving the shape and size of the objects. We will show that this specific problem class allows for a more compact and efficient formalization compared to the general case.

4 Formalization of Topological Relations

In principle, topological relations between simple polygons in the plane can be expressed using the x, y coordinates of all vertices of the two involved objects. However, while the approach has the advantage of being general enough to allow for all kinds of transformations to the objects during the optimization process, it requires a quadratic number of (in)equations. More precisely, for two convex polygons with m and n vertices, respectively, it requires $O(m \cdot n)$ (in)equations. In contrast, the approach described in the following employing Minkowski sums of the involved polygons requires only $O(m + n)$ (in)equations.

4.1 Minkowki-Based Formalization Approach

Given two polygons P and Q involved in a particular topological relation, the underlying idea of the Minkowski sum approach is to shrink one of the two polygons, let us say P, to a point p, while growing the other object Q accordingly. This growing is done using Minkowski sum operations and is, for instance, being employed in the area of motion planning (Latombe 1991) and layout generation (Marriott et al. 2001). We here extend this approach by defining two different Minkowski objects M^+ and M^- that allow for expressing aspects such as disjointness, containment and overlap between P and Q based on the relation between the point p and M^+ and M^-.

Given two point sets A and B, their Minkowski sum is defined as $M(A, B) = \{a + b \mid a \in A \wedge b \in B\}$. Instead of employing the Minkowski sum operation to P and Q directly, we replace P by the polygon we get from choosing a vertex p of P and taking $-(a - p)$ for each vertex a of P (basically translating P such that p lies at the origin and then point inflecting it on the origin). As illustrated in Fig. 3, we get

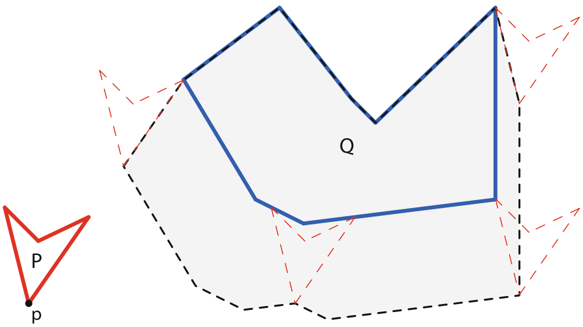


Fig. 3 Minkowski sum $M^+(p, P, Q)$ (*shaded area*) between two polygons P and Q

as a result the grown version of Q shown as the shaded region and referred to in the following as $M^+(p, P, Q)$:

$$M^+(p, P, Q) = \{-(a - p) + b \mid a \in P \wedge b \in Q\} \tag{1}$$

As indicated by the dashed versions of P, P will touch but not overlap Q exactly when p lies on the boundary of $M^+(p, P, Q)$. If p is outside of $M^+(p, P, Q)$, P and Q are disjoint. If P and Q are both convex, $M^+(p, P, Q)$ will be convex as well. Otherwise, it may be concave and have one or more polygonal holes.

To also be able to express containedness and formulate the complete set of topological constraints, we now introduce a second Minkowski sum $M^-(p, P, Q)$. While $M^+(p, P, Q)$ was defined using all points from P and Q, $M^-(p, P, Q)$ is based on the same Minkowski sum but using only those points of Q that lie on the boundary of Q written as ∂Q. "We then remove this area from Q, followed by a closure (cl) operation:"

$$M^-(p, P, Q) = cl(\, Q \setminus \{-(a - p) + b \mid a \in P \wedge b \in \partial Q\}\,) \tag{2}$$

As shown in Fig. 4, the result is a set of polygonal holes (shaded) in Q (two in this case), which in the general case may be concave. As again indicated by the dashed versions of P, the following holds: if p lies on the boundary of one of these holes, P will touch Q from the inside; if p lies in the interior of one of the holes, P is completely contained in Q.

To keep things simply, we equate the point sets $M^-(p, P, Q)$ and $M^+(p, P, Q)$ with the respective polygonal objects that describe their boundaries in the following instead of introducing a new notation. Both $M^-(p, P, Q)$ and $M^+(p, P, Q)$ can be computed using the general operation of convoluting polygons as for instance provided by the CGAL computational geometry library.[1] In the general case where both P and Q may be concave, $M^-(p, P, Q)$ and $M^+(p, P, Q)$ may have multiple

[1] http://www.cgal.org/

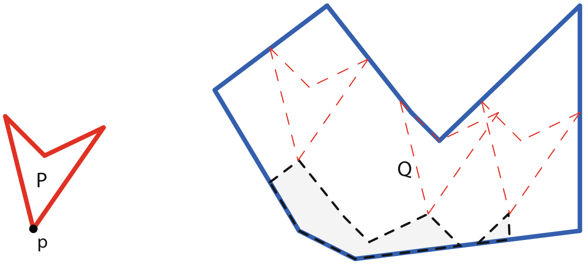


Fig. 4 Minkowski sum $M^{-}(p, P, Q)$ (*shaded area*) between two polygons P and Q

components and have holes which are themselves concave. We start our formalization of topological relations with the simplest case that P and Q (and hence also $M^{-}(p, P, Q)$ and $M^{+}(p, P, Q)$) are simple convex polygons and then show how to generalize the formalization to concave objects in Sect. 4.4.

4.2 Auxiliary Definitions

Point-to-Line Relations. We first define a set of auxiliary point-to-line relations for a point p and the directed line going through two other points q_1 and q_2. These will in the following be used to describe the position of a point with respect to the boundaries of a polygon and ultimately the topological relation between two polygons. We need to describe the position of p with respect to the line $\overrightarrow{q_1 q_2}$ in terms of being to the left, right, or on the line. This distinction can be made using the vector product of the two vectors $\overrightarrow{q_1 p}$ and $\overrightarrow{q_1 q_2}$. Depending on whether the result is $>, \geq, =, \leq, <$ to 0, the relation holding is r (right), *re* (right or equal), *eq* (equal), *le* (left or equal), or l (left).

$$eq(p, q_1, q_2) \Leftrightarrow (x_p - x_{q_1})(y_{q_2} - y_{q_1}) + (y_p - y_{q_1})(x_{q_2} - x_{q_1}) = 0 \tag{3}$$

$$l(p, q_1, q_2) \Leftrightarrow (x_p - x_{q_1})(y_{q_2} - y_{q_1}) + (y_p - y_{q_1})(x_{q_2} - x_{q_1}) < 0 \tag{4}$$

$$le(p, q_1, q_2) \Leftrightarrow (x_p - x_{q_1})(y_{q_2} - y_{q_1}) + (y_p - y_{q_1})(x_{q_2} - x_{q_1}) \leq 0 \tag{5}$$

$$r(p, q_1, q_2) \Leftrightarrow (x_p - x_{q_1})(y_{q_2} - y_{q_1}) + (y_p - y_{q_1})(x_{q_2} - x_{q_1}) > 0 \tag{6}$$

$$re(p, q_1, q_2) \Leftrightarrow (x_p - x_{q_1})(y_{q_2} - y_{q_1}) + (y_p - y_{q_1})(x_{q_2} - x_{q_1}) \geq 0 \tag{7}$$

Point-to-Polygon Relations. Assuming Q is a convex polygon with clockwisely ordered vertices q_1 to q_n, we now define a set of point-to-polygon relations $*\_cvx$. The first relation $boundary\_cvx(p, Q)$ holds if p lies on the boundary of Q. This is the case when it lies on one of the lines $\overrightarrow{q_i q_{i \oplus 1}}$ defined by the vertices of Q and is *re* to all such lines ($\oplus$ here stands for addition modulo n):

$$boundary\_cvx(p, Q) \Leftrightarrow \left(\bigvee_{i=1}^{n} eq(p, q_i, q_{i\oplus 1})\right) \wedge \left(\bigwedge_{i=1}^{n} re(p, q_i, q_{i\oplus 1})\right) \tag{8}$$

Point p is outside of Q if it is l of at least one of the lines $\overrightarrow{q_i q_{i\oplus 1}}$:

$$outside\_cvx(p, Q) \Leftrightarrow \bigvee_{i=1}^{n} l(p, q_i, q_{i\oplus 1}) \tag{9}$$

Similarly, we can define a predicate $outside\_boundary\_cvx(p, Q)$ as the disjunction of $boundary\_cvx(p, Q)$ and $outside\_cvx(p, Q)$ or, alternatively, using le instead of l in the previous equation. Point p is inside of Q (meaning it belongs to the interior of Q) if it is r of all edges:

$$inside\_cvx(p, Q) \Leftrightarrow \bigwedge_{i=1}^{n} r(p, q_i, q_{i\oplus 1}) \tag{10}$$

A predicate $inside\_boundary\_cvx(p, Q)$ ($boundary\_cvx(p, Q)$ or $inside\_cvx(p, Q)$) can be defined analogously. As we can see, all definitions so far consist of a number of (in)equations that grows linearly with the number of vertices of Q.

4.3 Topological Relations Between Convex Polygons

We now use the point-to-polygon relations from the previous section and the definitions of $M^+(p, P, Q)$ and $M^-(p, P, Q)$ from Sect. 4.1 to define the 11 topological relations (the eight RCC-8 base relations plus DR, PP, and PPI) between two convex polygons P and Q with vertices $p_1, p_2, \ldots p_m$ and $q_1, q_2 \ldots q_n$. p in the following definitions is always an arbitrarily chosen reference vertex of P. With both P and Q being convex, $M^+(p, P, Q)$ is always a single component convex polygon. $M^-(p, P, Q)$, on the other hand is either also a single convex polygon or does not yield a polygon at all as the corresponding Minkowski difference may be empty, a fact that we will write as $M^-(p, P, Q) = \emptyset$.

P is externally connected to Q (relation EC) if their boundaries touch but the interiors are disjoint. This is the case when p lies on the boundary of $M^+(p, P, Q)$. Hence, to formalize this relation, overall $2v^+$ (in)equations are needed where v^+ is the number of vertices in $M^+(p, P, Q)$ which is in $O(m + n)$.

$$EC\_cvx(P, Q) \Leftrightarrow boundary\_cvx(p, M^+(p, P, Q)) \tag{11}$$

P is disconnected from Q (relation DC) if there is no overlap at all between the objects. This is the case when p is outside of $M^+(p, P, Q)$, resulting in v^+ required inequations.

$$DC\_cvx(P, Q) \Leftrightarrow outside\_cvx(p, M^+(p, P, Q)) \tag{12}$$

P is DR of Q (the union of DC and EC) if p is *outside_boundary_cvx* of $M^+(p, P, Q)$ requiring again v^+ inequations.

$$DR\_cvx(P, Q) \Leftrightarrow outside\_boundary\_cvx(p, M^+(p, P, Q)) \tag{13}$$

Analogously, we can define TPP, NTPP, and their union PP using $M^-(p, P, Q)$. P is a tangential proper part of Q (relation TPP) if P is contained in Q and the boundaries touch. This holds true when $M^-(p, P, Q)$ is not empty and p lies on its boundary. $2v^-$ (in)equations are required where v^- is the number of vertices in $M^-(p, P, Q)$.

$$TPP\_cvx(P, Q) \Leftrightarrow M^-(p, P, Q) \neq \emptyset \wedge boundary\_cvx(p, M^-(p, P, Q)) \tag{14}$$

P is a *non*-tangential proper part of Q (relation NTPP) if in addition there is no overlap between the boundaries. This is the case when p is inside $M^-(p, P, Q)$ (taking v^- inequations).

$$NTPP\_cvx(P, Q) \Leftrightarrow M^-(p, P, Q) \neq \emptyset \wedge inside\_cvx(p, M^-(p, P, Q)) \tag{15}$$

Analogously,

$$PP\_cvx(P, Q) \Leftrightarrow M^-(p, P, Q) \neq \emptyset \wedge inside\_boundary\_cvx(p, M^-(p, P, Q)) \tag{16}$$

which again requires v^- inequations. The inverse relations TPPI, NTPPI, and PPI can simply be described by swapping P and Q, i.e.,

$$TPPI\_cvx(P, Q) \Leftrightarrow TPP\_cvx(Q, P) \tag{17}$$
$$NTPPI\_cvx(P, Q) \Leftrightarrow NTPP\_cvx(Q, P) \tag{18}$$
$$PPI\_cvx(P, Q) \Leftrightarrow PP\_cvx(Q, P) \tag{19}$$

To define partially overlap (relation PO), which means the interiors of the objects overlap but neither is contained in the other, both $M^+(p, P, Q)$ and $M^-(p, P, Q)$ are needed. PO holds if point p is at the same time inside $M^+(p, P, Q)$ and outside of $M^-(p, P, Q)$. Overall, $v^+ + v^-$ inequations are required to formalize this.

$$\begin{aligned} PO\_cvx(P, Q) \Leftrightarrow\ & inside\_cvx(p, M^+(p, P, Q)) \\ & \wedge \big(M^-(p, P, Q) = \emptyset \vee outside\_cvx(p, M^-(p, P, Q))\big) \end{aligned} \tag{20}$$

The equal relation EQ is a special case which most likely will not play a very important role in practice. We provide a simple formalization which is based on the following reasoning: for P and Q being in relation EQ, it is a prerequisite that they have the same number of vertices, same shape and size, and they need to be transformable into each other merely using displacement. If this is the case, the only requirement that needs to be checked during the adjustment is whether two previously determined corresponding vertices p of P and q of Q are equal:

$$EQ(P, Q) \Leftrightarrow p = q \tag{21}$$

The check whether EQ is possible at all and identification of corresponding vertices then are performed in a preprocessing step before the actual adjustment procedure, together with the computation of the different Minkowski polygons.

4.4 Generalization to Non-Convex Polygons

To generalize the previous formalization of topological constraints from convex polygons to non-convex polygons, we have to take into account that $M^+(p, P, Q)$ may now consist of a single component polygon as outer boundary written as $O_1(M^+(p, P, Q))$ which now may be concave, and in addition may have one or more polygonal holes $H_i(M^+(p, P, Q))$. $M^-(p, P, Q)$, on the other hand, can consist of several potentially concave components $O_i(M^-(p, P, Q))$ as in Fig. 4, but all of them without holes. As the outer boundaries and holes may all be concave themselves, we assume that they have been decomposed into convex components adding another index j to $O_1(M^+(p, P, Q))$, $O_i(M^-(p, P, Q))$, and $H_i(M^+(p, P, Q))$, e.g., $H_{i,j}(M^+(p, P, Q))$ for the jth convex polygon of $H_i(M^+(p, P, Q))$. Due to space limitations we here only provide two examples of the generalized definitions, namely those for EC and PO. The other relations are formalized analogously.

P is in relation EC with Q if p satisfies relation *boundary_cvx'* with one of the convex components of $O_1(M^+(p, P, Q))$ or one of the convex components of a hole $H_i(M^+(p, P, Q))$. *boundary_cvx'* here is slightly modified version of Eq. 8 in which the new boundaries resulting from the decomposition are excluded from the disjunction forming the left part of the definition in Eq. 8.

$$\begin{aligned} EC(P, Q) \Leftrightarrow & \left(\bigvee_j \mathit{boundary\_cvx}'(p, O_{1,j}(M^+(p, P, Q))) \right) \\ & \vee \left(\bigvee_j \mathit{boundary\_cvx}'(p, H_{i,j}(M^+(p, P, Q))) \right) \end{aligned} \tag{22}$$

For PO, p has to be inside one component of M^+ and simultaneously outside of all convex components of holes of M^+ and convex components of M^-.

$$\begin{aligned} PO(P, Q) \Leftrightarrow & \left(\bigvee_j \mathit{inside\_cvx}'(p, O_{1,j}(M^+(p, P, Q))) \right) \\ & \wedge \left(\bigwedge_{i,j} \mathit{outside\_cvx}(p, H_{i,j}(M^+(p, P, Q))) \right) \\ & \wedge \left(\bigwedge_{i,j} \mathit{outside\_cvx}(p, O_{i,j}(M^-(p, P, Q))) \right) \end{aligned} \tag{23}$$

The predicate *inside_cvx*′, in this case, is a modified version of Eq. 10 which only demands the relation *re* instead of *r* for boundaries introduced through the decomposition.

While, as these two examples illustrate, the definitions get significantly more complex in the general case, the number of (in)equations is again linear in the overall number of vertices in M^+ and M^-.

5 Adjustment Approach Using MNLP

Our approach for solving qualitative adjustment problems over topological relations is to transform them into mixed-integer nonlinear programming (MNLP) problems using the formalizations from the previous section. This MNLP formulation can then be fed into an MNLP solver and the returned solution can be retranslated into the final geometric solution $\mathcal{O}'$. If the MNLP was able to find an optimal solution to the MNLP problem, $\mathcal{O}'$ is guaranteed to be an optimal solution of the original qualitative adjustment problem. However, due to the heuristic nature of MNLP solvers which may get stuck in a local optimum, there is often no overall guarantee that an optimal solution is found in which case a better result may be achieved by finding a more compact or simplified formulations of the problem that avoids unnecessary variables and (in)equations.

MNLP problems have the following form: given

1. a vector $\boldsymbol{x} = (x_i)$ of variables over the reals,
2. a vector $\boldsymbol{y} = (y_j)$ of integer variables,
3. a nonlinear cost function $c(\boldsymbol{x}, \boldsymbol{y})$,
4. and a nonlinear constraint function $g(\boldsymbol{x}, \boldsymbol{y})$,

minimize/maximize $c(\boldsymbol{x}, \boldsymbol{y})$ subject to $g(\boldsymbol{x}, \boldsymbol{y}) \leq 0$.

We formalized all topological constraints using conjunctively and disjunctively combined sets of (in)equations. The involved real variables stand for the coordinates

Algorithm 1 Topological adjustment algorithm

```
function ADJUST($\mathcal{C}, N, \mathcal{O}$)
Input:
$\mathcal{C}$ qualitative calculus
$N$ qualitative constraint network over $\mathcal{C}$
$\mathcal{O}$ set of polygons $O_i$
Output:
$\mathcal{O}'$ new set of polygons $O'_i$ such that $O'_i$ is a displaced version of $O_i$
    choose reference vertices $p_i$ for each $O_i$ in $\mathcal{O}$
    compute $M^+(p_i, O_i, O_j)$ and $M^-(p_i, O_i, O_j)$ (if needed for their relation in $N$)
    compute convex decompositions of $M^+(p_i, O_i, O_j)$ and $M^-(p_i, O_i, O_j)$
    translate relations in $N$ into (in)equations over the coordinates of the $p_i$
    translate (in)equations into an equivalent MNLP problem
    run MNLP solver on MNLP problem resulting in modified reference points $p'_i$
    Generate $\mathcal{O}'$ by translating each $O_i$ by $p'_i - p_i$
    return $\mathcal{O}'$
end function
```

of the vertices. Since we are only considering displacement, it is sufficient in our formalization to choose one reference vertex for each input polygon and represent it by two variables, one for the x coordinate and one for the y coordinate. All other vertices occurring in the Minkowski polygons can be expressed relative to the chosen reference vertex. Hence, vector $\boldsymbol{x}$ consists of $2 \times n$ variables when n objects are involved. The mixed-integer approach is required to express disjunctions of (in)equations, which is not possible in the basic framework of simple (non)linear programming. As we will see in the following, introducing additional binary variables and additional inequations allows us to formulate the problem as a set of conjunctively connected inequations forming the constraint matrix $g(\boldsymbol{x}, \boldsymbol{y})$. The sequence of steps performed by the resulting overall adjustment algorithm is sketched in Algorithm 1.

5.1 Translation into an MNLP Problem

When translating the (in)equations from a qualitative adjustment problem into an MNLP problem, the coordinates of the reference points become the real variables x_i and initial values are specified corresponding to their original coordinates. Moreover, we introduce boundary constraints for each variable which restrict them to be within a certain range (e.g., $x_i \geq 0$ and $x_i < 1{,}000$).

The next step is to replace strict inequations ($<, >$) by $\leq$ and $\geq$, respectively, because MNLP does not allow for strict inequalities as they occur in the definitions of $l(p, q_1, q_2)$ and $r(p, q_1, q_2)$. To achieve this, a small ϵ value is added to one side of the equation. For instance, the equation $x_i < x_j$ is replaced by $x_i + \epsilon \leq x_j$. In our context, this ϵ value can also be used to assert a minimal level of disjointness, containedness, or overlap as it essentially corresponds to a buffer operation applied to the respective polygon.

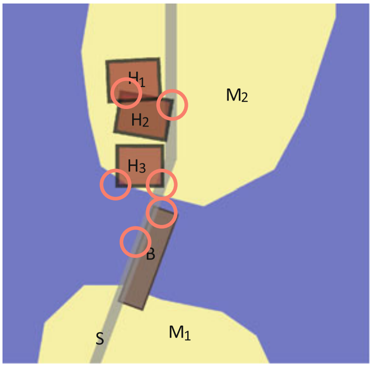
Next, disjunctions are resolved by introducing a binary integer variable $y_i \in \{0, 1\}$ into each (in)equation that is part of the disjunction such that it is always satisfied if y_i is 1. With the mentioned boundary constraints, it is easy to do this by inserting additive or subtractive terms consisting of a multiplication of y_i with a large constant C. For instance, assuming that all variables have been restricted to lie between 0 and $1,000$, the inequations in the disjunction $x_1 \geq x_2 \vee x_1 \leq x_3$ would be expressed as $x_1 + y_1\, C \geq x_2$ and $x_1 - y_2\, C \leq x_3$, respectively, with C being larger than 1,000. We then realize the actual disjunction via the additional constraint $y_1 + y_2 \leq 1$ (or in general $\sum_{i=1}^{n} y_i \leq n - 1$ for disjunctions of n (in)equations) which enforces that at least one of the y_i for the given disjunction is 0 and, hence, the original inequation has to be satisfied.

This approach adds n binary variables for a disjunction of n (in)equations and one additional inequation. The number of (in)equations per relation stays linear wrt. the number of vertices occurring in the Minkowski polygons, meaning a linear number of (in)equations is needed to describe one relation and, hence, $O(k \cdot r)$ (in)equations overall if there are r constraints in the input QCN and the maximum number of vertices over all Minkowski polygons is k. Given the input set contains m objects, the MNLP program will in addition contain $2 \times m$ real variables, $O(k \cdot r)$ binary variables, and $4 \times m$ boundary constraints. When the problem at hand requires a relational constraint between each pair of objects, the resulting $O(m^2)$ constraints would lead to $O(km^2)$ (in)equations when each relation is translated individually. This number could be reduced by exploiting the properties of the relations. For instance, if object A should be a NTPP of B and B should be in relation DC with C, it follows directly that A has to be in relation DC with C and, hence, the constraint does not need to be formalized at all.

To complete the MNLP, one still has to specify the cost function c. In our case, the costs only depend on the resulting values of the real variables x_i, not the binary integer variables. Taking a least square adjustment approach, we want to minimize the sum of the squared deviations of each variable from its original value. This corresponds to defining in the cost function $c(O_i, O_i')$ in the definition of the adjustment problem (see Definition 1) as being the squared displacement distance between the two geometries. Hence, given that the original value for each x_i is $\bar{x_i}$

$$c(x) = \sum_{i=1}^{n} (x_i - \bar{x_i})^2 \tag{24}$$

While this basic cost function is suitable for situations in which all objects are treated equally, more complex cost functions could be employed. For instance, preferences regarding which objects should be moved can be realized by introducing weights into the cost function.



$H_1\{\mathrm{DC}\}H_2,\ H_1\{\mathrm{DC}\}H_3,\ H_2\{\mathrm{DC}\}H_3$

$M_1\{\mathrm{DC}\}M_2$

$H_1\{\mathrm{NTPP}\}M_2,\ H_2\{\mathrm{NTPP}\}M_2,\ H_3\{\mathrm{NTPP}\}M_2$

$H_1\{\mathrm{DC}\}S,\ H_2\{\mathrm{DC}\}S,\ H_3\{\mathrm{DC}\}S$

$B\{\mathrm{PO}\}M_1,\ B\{\mathrm{PO}\}M_2$

$H_1\{\mathrm{DC}\}B,\ H_2\{\mathrm{DC}\}B,\ H_3\{\mathrm{DC}\}B$

$S\{\mathrm{DC}\}W$

$S\{\mathrm{PO}\}M_1,\ S\{\mathrm{PO}\}M_2$

Fig. 5 A geometric configuration and the topological integrity constraints that are supposed to be satisfied by the objects. Constraint violations are marked by the *circles*

6 Data Cleaning Experiment

Figures 5 and 6 show an application of our adjustment algorithm in a small data cleaning scenario. Data from different sources has been combined into a single spatial database. The data contains geometries for three houses (H_i), two mainland areas (M_i), a street (S) connecting both mainland areas, and a bridge (B) between them. As indicated by the circles in Fig. 5, there are several cases in which the combined data violates typical topological integrity constraints, for instance houses overlapping each other, houses overlapping the street, or houses not being completely located on land. Similarly, the bridge does not connect the two mainland areas and the street goes partially through the water. On the right side, we see the text representation of a QCN stemming from the application of typically integrity constraints to this scenario. It, for instance, demands that the bridge overlaps both mainland areas, that the street is DC from the water (W)[2], that the houses have to be NTPP of M_2 and be in relation DC to each other, etc. We applied our topological adjustment algorithm to this input problem using the mentioned CGAL library for computational geometry operations such as computing the Minkowski polygons and Bonmin[3] as the MNLP solver. The ϵ parameter was chosen such that a small minimal distance is enforced for relations DC, NTPP, and PO.

The solution computed by our approach is shown in Fig. 6 (left). The adjustment procedure was able to indeed resolve all constraint violations in the input data. The resulting MNLP problem contained 18 real variables, 162 binary variables, and 290 inequations. The adjustment procedure ran for 98.7 s on a 3 GHz i5 computer. To illustrate the performed displacements, Fig. 6 (right) shows an overlay of the original configuration and the adjusted configuration. While certain requirements

[2] This constraint needed a special treatment when modeling this scenario because the water does not exist as an adjustable input object but rather forms the background.

[3] http://www.coin-or.org/Bonmin/

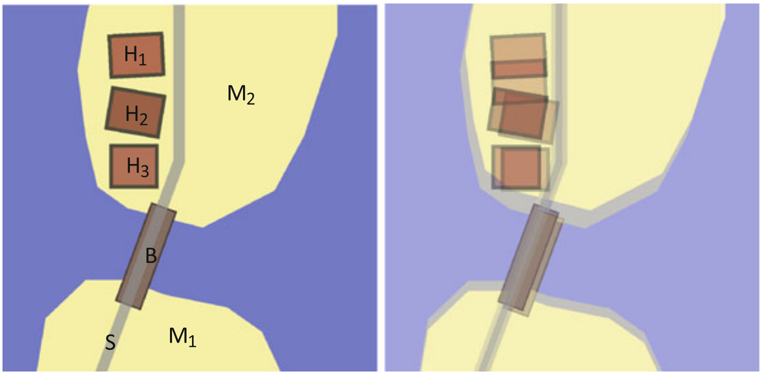


Fig. 6 Result of the adjustment procedure (*left*) and an overlay of the initial configuration and the result illustrating the displacements (*right*)

could undoubtedly have been chosen differently and, additionally, one could have required that certain objects should not be allowed to be moved at all (e.g., the mainland areas), this example nevertheless demonstrates the ability of our approach to successfully cope with the full set of topological relations.

7 Conclusions

We described an approach for solving the qualitative adjustment problem of adapting geometric data to satisfy a set of qualitative spatial integrity constraints for the specific case of polygonal objects in 2D which are supposed to satisfy certain topological relations and only allowed to be displaced. We showed that for convex polygons, a general formalization of topological constraints requiring only a linear number of (in)equations per relation is possible based on the notion of Minkowski sums. In our overall approach—which we expect to be applicable to other qualitative calculi and other transformations as well—we transformed the formalization into an equivalent MNLP problem which is then solved using a dedicated MNLP solver.

While initial experiments indicate that the approach is able to solve adjustment problems over the complete set of RCC-8 relations, future research will aim at improving the efficiency and expressivity of our approach. For instance, we intend to employ QSR techniques to improve the performance by removing redundant constraints and (in)equations, before feeding the problem into the MNLP solver, as well as to decompose large problems into independently solvable subproblems. To increase the expressivity, we plan to provide means for defining constraints about newly constructed entities, e.g., the intersection or union of input objects, parts of entities, as well as to combine the approach with other kinds of constraints, both qualitative and quantitative in nature. Finally, we will investigate to what extent the ideas underlying our approach are applicable when other transformations are allowed. For instance, we will attempt to integrate the Minkowski computation into the actual adjustment process to allow for free deformation by adjusting the positions of individual vertices.

Acknowledgments The author would like to thank Reinhard Moratz and the anonymous reviewers for valuable feedback and suggestions.

References

Bhatt M, Lee JH, Schultz C (2011) CLP(QS): a declarative spatial reasoning framework. In: Egenhofer MJ, Giudice NA, Moratz R, Worboys MF (eds) Proceedings of the 10th international conference on spatial information theory (COSIT). Lecture notes in computer science, vol 6899. Springer, pp 210–230

Clementini E, Sharma J, Egenhofer MJ (1994) Modeling topological spatial relations: strategies for query processing. Comput Graph 18(92):815–822

Cohn AG, Hazarika SM (2001) Qualitative spatial representation and reasoning: an overview. Fundamenta Informaticae 46(1–2):1–29

Egenhofer MJ (1991) Reasoning about binary topological relations. In: Gunther O, Schek HJ (eds) Advances in spatial databases, second symposium on large spatial databases, vol 525. Springer, New York, pp 143–160

Haarslev V, Möller R (1997) Spatioterminological reasoning: subsumption based on geometrical inferences. In: Brachman RJ, Donini FM, Franconi E, Horrocks I, Levy AY, Rousset MC (eds) Description logics, vol 410, URA-CNRS (1997)

Harrie L (1999) The constraint method for solving spatial conflicts in cartographic generalization. Cartogr Geogr Inf Sci 26(1):55–69

Klippel A, Worboys M, Duckham M (2008) Identifying factors of geographic event conceptualisation. Int J Geogr Inf Sci 22(2):183–204

Lamy S, Ruas A, Demazeau Y, Jackson M, Mackaness W, Weibel R (1999) The application of agents in automated map generalisation. In: 19th international cartographic conference, pp 160–169

Latombe JC (1991) Robot motion planning. Kluwer Academic Publishers, Boston

Marriott K, Moulder P, Stuckey P, Borning A (2001) Solving disjunctive constraints for interactive graphical applications. In: Walsh T (ed) Principles and practice of constraint programming, vol 2239. Springer, Berlin, pp 361–376

Randell DA, Cui Z, Cohn A (1992) A spatial logic based on regions and connection. In: Proceedings of the third international conference on principles of knowledge representation and reasoning. Morgan Kaufmann, pp 165–176

Renz J, Nebel B (2007) Qualitative spatial reasoning using constraint calculi. In: Aiello M, Pratt-Hartmann IE, van Benthem JF (eds) Handbook of spatial logics. Springer, Heidelberg, pp 161–215

Sester M (2000) Generalization based on least squares adjustment. In: International archives of photogrammetry and remote sensing, vol 33

Steiniger S, Weibel R (2007) Relations among map objects in cartographic generalization. Cartogr Geogr Inf Sci 34(3):175–197

Ware JM, Jones CB (1998) Conflict reduction in map generalization using iterative improvement. GeoInformatica 2(4):383–407

Automatic Map Retrieval and Map Interpretation in the Internet

Volker Walter, Fen Luo and Dieter Fritsch

Abstract The Internet contains huge amounts of maps representing almost every part of the Earth in many different scales and map types. However, this enormous quantity of information is completely unstructured and it is very difficult to find a map of a specific area and with certain content, because the map content is not accessible by search engines in the same way as web pages. However, searching with search engines is at the moment the most effective way to retrieve information in the Internet and without search engines most information would not be findable. In order to overcome this problem, methods are needed to search automatically for maps in the Internet and to make the implicit information of maps explicit so that machines can process it. In this paper we discuss how maps can be found automatically in the Internet and moreover, how the content of maps can be interpreted automatically.

Keywords Interpretation · Data mining · Internet · Retrieval · Databases

1 Introduction

Searching for cartographic maps in the Internet is nowadays only conditionally possible. Text search engines allow searching for text phrases in text documents. With this technology it is not possible to search for maps because maps are not represented

V. Walter (✉) · F. Luo · D. Fritsch
Institute for Photogrammetry, Stuttgart University, Geschwister-Scholl-Str. 24D,
70174 Stuttgart, Germany
e-mail: Volker.Walter@ifp.uni-stuttgart.de

F. Luo
e-mail: fenluo@hotmail.de

D. Fritsch
e-mail: Dieter.Fritsch@ifp.uni-stuttgart.de

S. Timpf and P. Laube (eds.), *Advances in Spatial Data Handling*,
Advances in Geographic Information Science, DOI: 10.1007/978-3-642-32316-4_14,

as text documents, but in raster (TIFF, GIF, JPEG, etc.) or vector formats (Shapefile, KML, GeoVRML, etc.).

In principle, maps in raster formats can be retrieved with image search engines. Image search engines allow for searching for images by evaluating text phrases. The result is a list of images that are embedded in a website that contains the phrase or which have a filename containing the phrase. However, the content of the images is not evaluated. Therefore, completely wrong images can be found and only those images will be found, that can be related with the phrase. Spatial queries (e.g. find all maps in a coordinate rectangle) or semantic queries (e.g. find all maps with golf courses and sea view) are with this approach not possible.

Some image search engines additionally provide the option to search for similar images (e.g. Google). The result is a list of images with similar appearances. The similarity is calculated from the image appearance with Content Based Image Retrieval (Datta et al. 2008) and from the text in the embedding website and is also not based on spatial or semantic similarity. Therefore, spatial and semantic queries are not possible.

Even if it would be possible to search automatically for images that represent maps, the subsequent map interpretation of raster maps is difficult, because the map objects are represented only implicitly with raster cells. Therefore, raster maps must first be segmented into map objects. This is a challenging task, because maps can have very different appearances. For this reason we use vector data as input for our process. Vector data consist of map objects with explicitly modeled geometries. Additionally, maps in vector formats can have semantic information (objects belong to object classes and objects can have semantic attributes), which can be used to support the interpretation process.

In the first part of the paper we describe how files, which contain spatial vector data, can be retrieved in the Internet. This is realized by combining the results of a text search engine with a web crawler. In the second part we describe the automatic interpretation of the map type by using Kohonen Feature Maps. The same technique can be used for the interpretation of map objects. This is described in Walter and Luo (2011).

2 Existing Work

The search for data, which cannot be retrieved by text search engines, is object of many scientific studies. However, most of them concentrate on the interpretation task and not on the retrieval, such as Funkhouser et al. (2003) who discuss the search for 3D models in the World Wide Web. A (simple) web crawler is used for retrieval to build a database. Since one web crawler can retrieve only a small percentage of all web pages, the resulting database contains only a small number of 3D models.

The performance of a web crawler can be improved with parallel crawling. Singh and Singh (2010) propose a parallel architecture in which a web crawler is subdivided into a central crawler, a central database, one or more crawl frontiers, and a local database for each crawl frontier.

Local search engines such as SPIRIT (Jones et al. 2004), Google Local (http://local.google.com) or Yahoo! Local Maps (http://maps.yahoo.com) provide spatial searches. They are based on map-and-hyperlink structures, which can be inputted manually or derived automatically from gazetteers by searching for place names in the text of Internet pages. The automatic retrieval of non-text information is with that technology not possible. However, gazetteers are an important information source for map interpretation (see below). Digital globes (e.g. Google Earth or Microsoft Bing) also provide spatial searches, but only based on data that is stored in their systems. External maps can only be related to this data, if users explicitly define them as overlays.

The Internet can be retrieved for spatial information that is not explicitly modeled with maps or geometrical objects. For example, Jones et al. (2008) discuss an approach to model and find the boundaries of vague places based on an evaluation of web pages. A vague place is a place with fuzzy boarders (e.g. South of France or Rocky Mountains). The assumption of this approach is that web pages, that contain the name of a vague place, contain also very often places with well-known coordinates. The extension of a vague place is then calculated based on a density surface modeling of the frequency of the occurrence of the co-located places.

Automatic map interpretation is a topic that has been discussed in different approaches. One criterion to differentiate between different map interpretation approaches is the type of input data (raster or vector data). A raster-based approach for the automatic interpretation of scanned topographic maps with query languages can be found in Graeff and Carosio (2002). The interpretation is done with pattern recognition algorithms in the raster domain. The detected objects are implicitly contained in the raster images but were explicitly modeled when the corresponding analogue map was produced. Therefore, the objects are already visible, but cannot be queried because of the raster representation. A raster-based approach for the automatic determination of settlement structures from vector road data sets can be found in Walter (2008).

A combination of a raster- and vector-based approach for automatic map interpretation is discussed in Viglino and Pierrot-Deseilligny (2003). The input for this process is a raster map that is converted into a vector representation. Different object classes (for example buildings, hangars or parcels) are reconstructed with low-level primitive extraction and subsequent classification. Vector approaches are often based on techniques from the field of Artificial Intelligence (AI). For example, Sester (2000) presents an approach for the semi-automatic interpretation of unstructured vector data based on machine learning techniques. A graph-based approach for clustering unstructured point data can be found in Anders (2003). The approach is completely parameter-free and can be applied to very different data sets.

The automatic derivation of unknown information from databases is also known under the term Data Mining or Knowledge Discovery (Frawley et al. 1991). In the context of spatial data, these techniques are also called Geographic Knowledge Discovery (GKD) or Geographic Data Mining (Miller and Han 2009). Data Mining techniques are used to derive unknown information from huge data sets that is not visible for a human. This applies only partly to this work, because we want also to derive information that is visible for humans, but which is not modeled and stored explicitly in the database. On the other hand the information can be visible, but people do not see it, because the information might be cluttered by other information, e.g. a roundabout is present in a road vector set but it might not be seen immediately, when it is not highlighted. Therefore, map interpretation can be seen as a mixture between data mining and image interpretation.

Data mining approaches in context of spatially aware search engines are discussed in Heinzle and Sester (2004). They describe the automatic extraction of classical metadata from spatial data sets and concepts of information retrieval to derive implicit information with data mining algorithms. In Heinzle et al. (2007) this work is continued and algorithms for the automatic recognition of patterns in road network data are developed. The search for patterns in maps in order to detect implicit information for the automatic map generalization is described in Mackaness and Edwards (2002). They argue that any map can be seen as a subset of possible patterns and a map generalization is a set of transformations from one pattern to another. An ontology driven pattern recognition approach for the detection of terraced houses in vector data is presented in Lüscher et al. (2008). They use ontologies to describe the characteristics of terraced houses and map this ontology onto a pattern recognition process.

Steinhauer et al. (2001) present a method for the automatic interpretation of abstract regions in a map. An abstract region consists of several map objects, which are grouped to a single object. The process is subdivided into two steps. First, region candidates are selected based on an evaluation of neighborhood relations. Then, objects, which consist of a hierarchical combination of single objects, are recognized with a grammar-based compiler approach.

The interpretation of spatial data cannot only be done with 2D data but also with 3D data. For example, Schleinkofer (2007) uses Neural Networks in order to classify building constructions. The construction elements are classified into the categories walls, doors, windows, ceilings, ceiling openings, pillars and beams by an evaluation of their spatial extension, surface area, intersection area with other objects and object coordinates.

Automatic sketch interpretation is a problem, which has many similarities to the problem of automatic map interpretation (Wuersch and Egenhofer 2008). However, in sketch interpretation the main focus is more on segmentation, classification and labeling (Sezgin and Davis 2005), whereas in map interpretation the focus is more on the following tasks, like clustering or data mining. Also the abstraction level of the input data in sketch interpretation is typically higher as in map interpretation. Maps consist typically of well-formed geometrical objects whereas sketches could also be represented by very simple geometrical entities. Nevertheless, both research areas have a large overlap.

3 Web Crawling

In a first step, we search in the Internet for files that contain spatial vector data. The search for specific file types is only restricted possible with existing search engines. Many search engines do not support the search for specific file types at all, such as Microsoft Bing or Lycos. Other search machines support the search for specific file types, but only for a limited set of file types. For example, Google supports the search for the file types: pdf, ps, dwf, kml, kmz, xls, ppt, rtf and swf. Although the file types kml and kmz represent geographical features, in most of all cases they contain only the coordinates of points of interest and not comprehensive map data. In contrast, Esri Shapefile (shp) is a very popular geospatial vector data format for geographical information and a huge amount of maps in shape format are available in the Internet.

Since commercial search engines do not support the search for Shapefiles, we developed a web crawler for this task. A web crawler is a computer program that browses the World Wide Web in a systematical way. A web crawler starts at a predefined web page and extracts all links of this page. Then, the web crawler follows the links and again extracts the links of the linked web pages. This is repeated until a break criterion is reached or the whole World Wide Web is retrieved. The visited pages are stored in a database to avoid that a link that has already been followed is used again.

Different strategies can be used to optimize the search result of a web crawler: depth-first search, breadth-first search and best-first search (see Fig. 1). In depth-first search, the web crawler starts at a predefined page, extracts the links of this page and follows the first link. Again the links are extracted and the first link is followed. This is repeated until no new link can be found. The next link that is used is the second link of the first page. In breath-first search, the web crawler also starts at a predefined page and extracts the links of this page. Each link is followed and all links of the next level are extracted. These links are again followed and all links of the next level are extracted, etc. In best-first search, the links are ranked according to a measure, which quantifies the relevance of the links. With this strategy it is possible to find relevant pages faster, but the definition of an appropriate measure is often very difficult.

Since the World Wide Web contains an enormous amount of pages, it is not possible to retrieve the whole Web with one single web crawler. For this reason we have developed an alternative strategy to decrease the search space. First, we search for a specific textual search term (for example: “Shapefile download”) with Google. Then, the web crawler retrieves only the web pages of the corresponding result list. The web pages are retrieved with a breadth-first search, which evaluates only the first three link levels, since we assume that the web page contains a direct link to a Shapefile or an indirect link, which can be accessed by following maximum two links. Additionally we evaluate maximum 30,000 links at one server. This avoids that web servers with a huge amount of web pages are completely evaluated, such as Wikipedia. Since Wikipedia is a very popular web site, web pages of Wikipedia are very often at the top in the result list of a Google search.

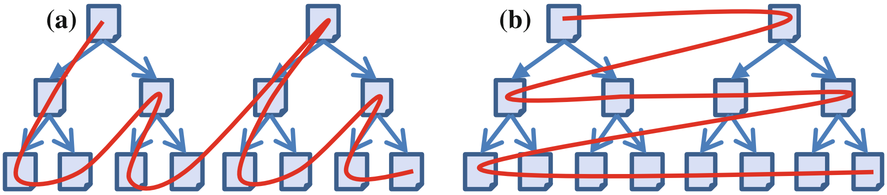


Fig. 1 Depth-first search (**a**) and breadth-first search (**b**)

Table 1 Results of different search strategies

Strategy	Number of visited servers	zip files	shp files	hit rate (%)
Breadth-search	9	23	0	0,0
“Shapefile download”	33	25.188	4.594	1,53
“Shapefile free”	18	12.264	629	0,20
“Shapefile”	14	2.992	528	0,18

Shapefiles in the Internet can be found normally only in zip-achieves, since the information of an ArcGIS geodatabase is normally stored in different files, which must be used together (e.g. shp-file contains the geometrical data, dbf-file contains the thematic data, shx-file contains a positional index and prj-file contains coordinate system and projection information). Therefore the web crawler searches for zip-files, extracts the content of the zip-file and then searches for Shapefiles.

We tested our approach with different configurations: (1) a normal breath-search without any limitations and without using a Google result list (the entry point of the web crawler was the homepage of the Institute for Photogrammetry: www.ifp.uni-stuttgart.de) and (2)–(4) with the described strategy and using a Google result list with the search terms (2) “Shapefile download”, (3) “Shapefile free” and (4) “Shapefile”. The web crawler retrieved exactly 300.000 web pages for all strategies. Table 1 shows the results of the different searches.

The combination of a web crawler with a Google search increases considerably the hit rate. The search for Shapefiles with only one single web crawler without using an intelligent search strategy is not efficient.

In the next step we examined if the result lists of the searches with different search terms overlap. Figure 2 shows an evaluation of the overlap of the hit lists.

The overlap of the hit lists of the final result depends mainly on the overlap of the corresponding Google hit lists. Overlapping results can be avoided if the different search terms are combined in one Google search by combining them with “OR”. The combined search term for all three web searches is *“Shapefile free” OR “Shapefile download” OR “Shapefile”.*

The term “Shapefile” can be combined also with other words like “Shapefile gratis”, “Shapefile database” or “Shapefile server”. In future research we want to investigate how we can find automatically optimized search term combinations.

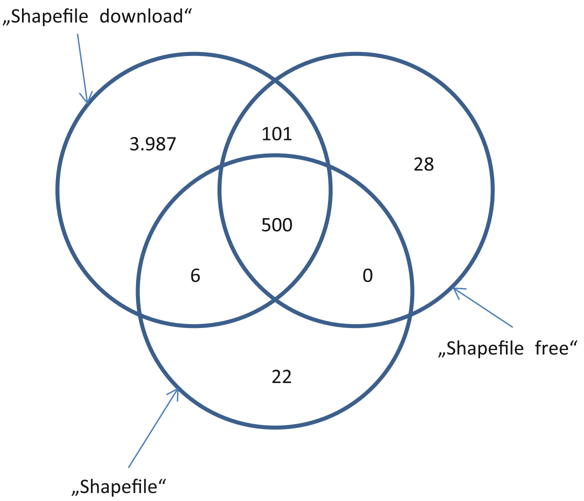


Fig. 2 Overlap of the hit lists of different search strategies

4 Map Interpretation

Automatic map interpretation approaches can be subdivided into approaches that derive metadata (information about the map: map type, map extension, number of map elements, etc.) and approaches that interpret the map content (interpretation of objects, grouping of objects, region interpretation etc.). In the following we discuss how the map type can be interpreted. An approach for the interpretation of map objects can be found in Walter and Luo (2011).

For the interpretation of the map type we use a Kohonen Feature Map, which is a special type of Artificial Neural Networks. Artificial Neural Networks are an approach to simulate biological neural networks. Different types of Artificial Neural Networks have been developed, like single-layer feed-forward networks, multilayer-networks, recurrent networks or Self-Organizing Maps. The Kohonen Feature Map was developed by Tuevo Kohonen (1982) and is a type of a Self-Organizing Map, which uses unsupervised learning in order to organize the connections between the neurons. That means that the neural network does not know what the correct classification for a specific input is. The function of the network is therefore to categorize different inputs into clusters, which have similar characteristics. Besides Kohonen Feature Maps there exist other unsupervised learning techniques like k-means, competitive learning or vector quantization. In principle every unsupervised learning technique can be used for map interpretation. We use Kohonen Feature Maps because they are easy to implement, they can be trained without much effort and they are robust against noisy input data. The following realization of a Kohonen Feature Map is based on a self-developed Java program.

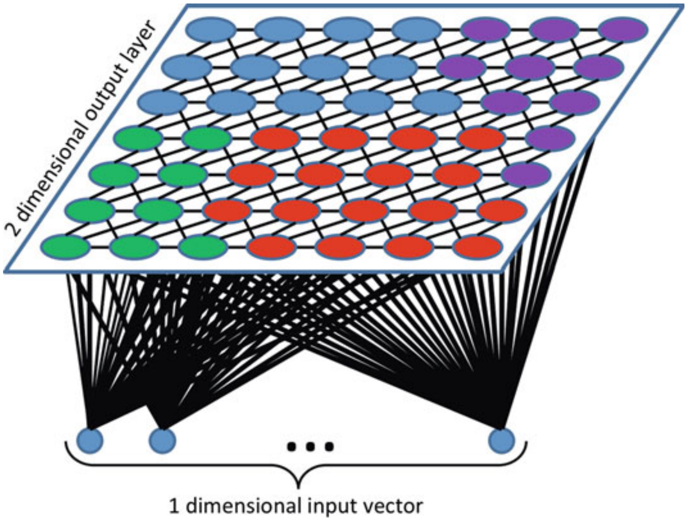


Fig. 3 Topology of a Kohonen Feature Map

A Kohonen Feature Map consists of an input layer and an output layer (see Fig. 3). The layers are represented with neurons. Objects, that should be classified, are represented with a feature vector, which describes the object characteristics. This feature vector is the input for the Kohonen Feature Map. The definition of the characteristics of the feature vector is the most important part and has to be done very carefully. The output is a classification of the objects into different object classes. The classification consists of a learning phase and a mapping phase.

Each neuron of the output layer corresponds to an object type and is facilitated or inhibited by the other neurons. Each object type has a center of activation on the output layer, which represents the most appropriate neuron of this object type. Around the center of activation are other neurons, which correspond to the same object type, but with a decreasing activation the greater the distance to this center is. The search for the center of activation for each object type is the main part of the learning phase. A detailed description of the mathematical basics of Kohonen Feature Maps can be found in Agarwal and Skupin (2008).

In the following, the practical use of Kohonen Feature Maps for the classification of the map type is discussed. We tested the classification approach on 127 maps that we selected manually from the results of the automatic web search. The selection criteria was to use only those maps which can be interpreted by a human without problems because many of the found maps contained information which was hard or even impossible to interpret (even for a human). Figure 4 shows typical examples of six different map types that should be interpreted.

Figure 4a shows an example of a map that contains a grid that represents map sections. The sections are represented with a rectangular grid. Figure 4b shows an example of a map with building footprints. Figure 2c shows a map that represents

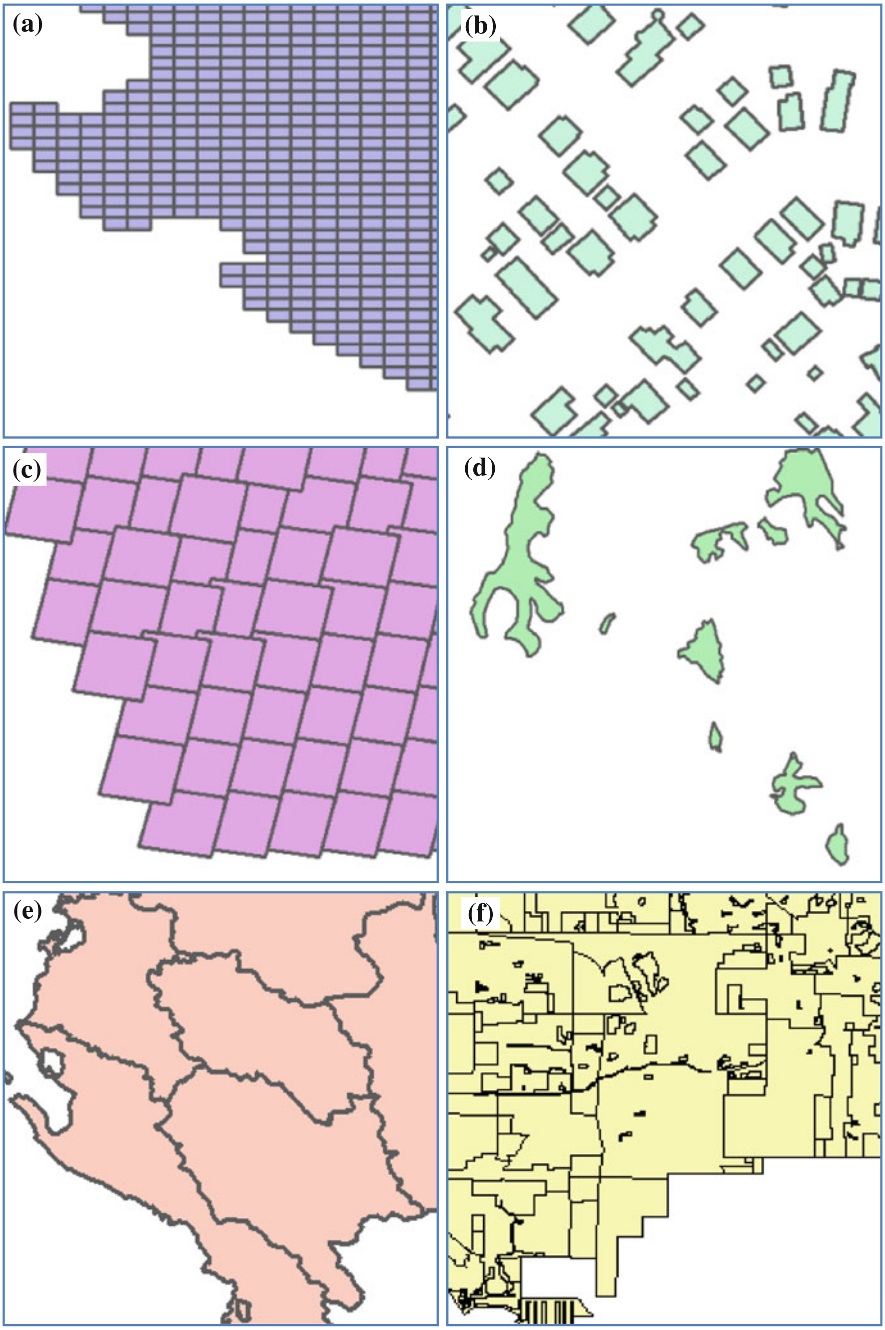


Fig. 4 Different map types which can be classified with the approach

Table 2 Characteristics for different map types

Object characteristics	Map grids	Buildings	Satellite images	Lakes	Regions	City maps
Regular area size	1	1	1	0	0	0
Irregular area size	0	0	0	1	1	1
Area size < 10.000 m^2	1	1	1	0	0	1
Area size > 10.000 m^2	0	0	0	1	1	0
With perpendicularity	1	1	0	0	0	1
Without perpendicularity	0	0	1	1	0	0
Neighbored polygons	1	0	1	0	1	1
Non-neighbored polygons	0	1	0	1	0	0
Surrounding polygons	0	0	0	0	0	1

footprints of Landsat TM images. The footprints are overlapping and are organized in a non-rectangular grid with grid cells that have a uniform size. This structure is also typical for other satellite images footprints. Figure 4d shows an example of a map that contains lakes. Figure 4e represents a map with regions. The characteristics of regions are that they are connected and not overlapping and have irregular shapes. Figure 4f shows a map that represents a city plan. City plans are similar to region maps with the difference that they contain more rectangular structures. Beside these six different map types there exist more map types that have typical appearances. In the following we evaluate only the described types but the approach can be straightforwardly extended to interpret more map types.

The characteristics of the different map types have to be defined with a vector consisting of 0 and 1 values. Altogether nine characteristics are used (see Table 2) which results in an input vector with the dimension 9.

An object characteristic in the table has the value “1” if 75 percent of all map objects fulfill the characteristic. In our case all map objects are represented with polygons. The characteristics are defined as:

- *Regular area size*: average size of all polygons/*abs*(size of polygon—average size of all polygons) $>$ 10
- *Irregular area size*: average size of all polygons/*abs*(size of polygon—average size of all polygons) <= 10
- *With perpendicularity*: polygon must have at least 3 angle with 85 ° $<$ angle $<$ 95 °
- *Without perpendicularity*: polygon has less than 3 angle with 85 ° $<$ angle $<$ 95 °
- *Neighbored polygons*: polygon has at least 2 points which are adjacent to another polygon
- *Non-neighbored polygons*: polygon has less than 2 points which are adjacent to another polygon
- *Surrounding polygons*: polygon is contained in another polygon

The selection of appropriate object characteristics is the most important part of the definition of a Kohonen Feature Map. Typically, a good configuration can only be

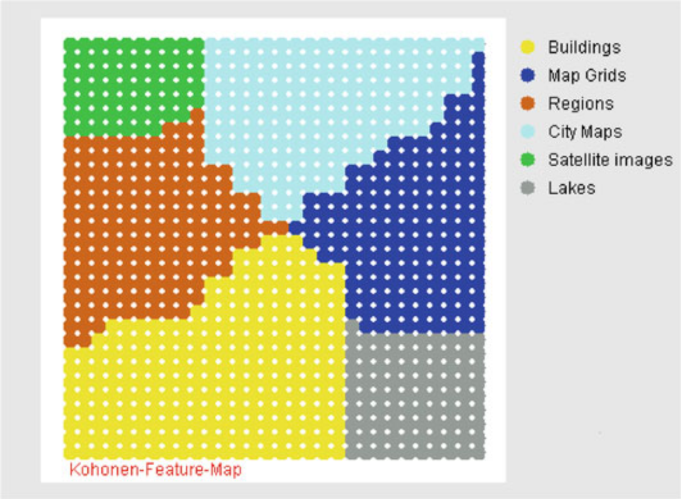


Fig. 5 Output layer of the Kohonen Feature Map after the training phase

Table 3 Results of the map type interpretation

	Map grids	Buildings	Satellite images	Lakes	Regions	City maps
Number of maps	7	7	6	70	29	8
Correct interpreted	7	7	6	60	24	8

found by testing and optimizing different combinations of possible characteristics. In the next step, the numbers of neurons of the Kohonen Feature Map have to be defined. The number of neurons of the input layer corresponds to the number of different object characteristics. The optimum number of neurons of the output layer was estimated by testing. Based on an evaluation of different configurations, we use 30*30 neurons in the output layer. After the training phase, the neurons of the output layer of the Kohonen Feature Map can be colored depending on the object type, which they are representing (see Fig. 5). The final results of the interpretation using this trained Kohonen Feature Map are shown in Table 3.

The results of the automatic interpretation of the map type are very good. The main reason for this is that the different map types have clear distinguishable characteristics. If we use more different map types, the definition of the object characteristics will be more overlapping and the number of wrong classified map types will presumably be higher. We will make more comprehensive tests to examine this situation.

Other future work will be to use other techniques for map interpretation. Additional information can be used in order to support the interpretation process. For instance, the map filename is very often a very helpful information source. Many maps have filenames like "Germany_Rivers.shp". The filename contains information about the area of the map and about the object types in the map. If a map is not geocoded, it can be tested if parts of the filename are contained in a gazetteer. A gazetteer is a geographical directory, which contains toponyms and their spatial extension. Also it can be checked if parts of the filename are contained in a geographical ontology to get information about the type of the map objects. The same techniques can be applied if the objects are structured in different layers or if they have semantic attributes. As input we can use the layer names or the attribute names.

5 Summary

The World Wide Web contains many vector maps in Shapefile format that can be used for spatial analyses. The problem is to find them since the number of existing web pages is enormous. According to http://www.worldwidewebsize.com/ the indexed Web contained at least 7.5 billion pages at the end of 2011. For this reason we developed an approach, which combines a textual Google search with a web crawler, to decrease the search space. With the proposed method it is possible to find with only one PC a considerably high number of Shapefiles in short time.

The next step is an automatic interpretation to make the implicit map information explicit. Map interpretation is a complex process that consists of different tasks. Beside the interpretation of the map type we need techniques for the interpretation of the map scale, map extension and map content. All tasks are dependent from each other. For example, it becomes easier to interpret the map type when the map scale is known and vice versa. The proposed method in this paper shows that we are already able to solve specific parts of automatic map interpretation, but a lot of research still has to be done in this field.

References

Agarwal A, Skupin A (eds) (2008) Self-organizing maps: applications in geographic information science. Wiley, West Sussex

Anders KH (2003) A hierarchical graph-clustering approach to find groups of objects. In: ICA commission on map generalization, technical paper at the fifth workshop on progress in automated map generalization, IGN, Paris, published on CD-ROM, Internet access: http://www.geo.unizh.ch/ICA/docs/paris2003/papers03.html. Accessed 5 Jan 2012

Datta R, Joshi D, Li J, Wang J (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2), Article 5, p 60

Frawley W, Piatetsky-Shapiro G, Matheus C (1991) Knowledge discovery in databases: an overview. In: Piatetsky-Shapiro G, Frawley W (eds) Knowledge discovery in databases. AAAI/MIT Press, Menlo Park, p 27

Funkhouser T, Min P, Kazhdan M, Chen J (2003) A search engine for 3D models. ACM Trans Graph 22(1):83–105

Graeff B, Carosio A (2002) Automatic interpretation of raster-based topographic maps by means of queries. FIG XXII international congress Washington, published on CD-ROM, Internet access: http://www.fig.net/pub/fig_2002/Ts3-10/TS3_10_graeff_carosio.pdf. Accessed 5 Jan 2012

Heinzle F, Sester M (2004) Derivation of implicit information from spatial data sets with data mining. Int Arch Photogrammetry Remote Sens 35(Part B4):335–340

Heinzle F, Anders KH, Sester M (2007) Automatic detection of pattern in road networks-methods and evaluation. In: Proceeding of joint workshop visualization and exploration of geospatial data, vol XXXVI-4/W45, published on CD-ROM, Internet access: http://tiny.cc/FEyb7. Accessed 5 Jan 2012

Jones JB, Abdelmoty AI, Finch D, Fu D, Vaid S (2004) The SPIRIT spatial search engine: architecture, ontologies and spatial indexing. In: Proceedings of Geographic information science: third international conference, GIScience 2004, Adelphi, MD, USA, 20–23 Oct 2004

Jones C, Purves R, Clough P, Joho H (2008) Modeling vague places with knowledge from the Web. IJGIS 22(10):1045–1065

Kohonen T (1982) Clustering, taxonomy, and topological maps of patterns. In: Proceedings of international conference on pattern recognition (ICPR), Washington, IEEE Computer Soc. Press, pp 114–128

Lüscher P, Weibel R, Mackaness A (2008) Where is the terraced house? On the use of ontologies for recognition of urban concepts in cartographic databases. In: Ruas A, Christopher G (eds) Headway in spatial data handling. Lecture notes in geoinformation and cartography, Springer, Berlin, pp 449–466

Mackaness W, Edwards G (2002) The importance of modeling pattern and structure in automated map generalisation. In: Proceedings of joint workshop on multi-scale representations of spatial data, Ottawa, Canada, published on CD-ROM, Internet access: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.1311\&rep=rep1\&type=pdf. Accessed 5 Jan 2012

Miller HJ, Han J (eds) (2009) Geographic data mining and knowledge discovery, 2nd edn. Taylor and Francis, New York

Schleinkofer MF (2007) Wissensbasierte Unterstützung zur Erstellung von Produktmodellen im Baubestand. Technische Universität München, Dissertation

Sester M (2000) Knowledge acquisition for the automatic interpretation of spatial data. Int J Geog Inf Sci 14(1):1–24

Sezgin TM, Davis R (2005) HMM-based efficient sketch recognition. In: Proceedings of the international conference on intelligent user interfaces (IUI'05), ACM Press, pp 281–283

Singh A, Singh K (2010) Faster and efficient web crawling with parallel migrating web crawler. IJCSI Int J Comput Sci, Issues 7(3), no. 11:28–32

Steinhauer JH, Wiese T, Freksa C, Barkowsky T (2001) Recognition of abstract regions in cartographic maps. In: Proceedings of the international conference on spatial information theory: foundations of geographic information science, pp 306–321

Viglino JM, Pierrot-Deseilligny M (2003) A vector approach for automatic interpretation of the french cadastral map. In: Proceedings of the seventh international conference on document analysis and recognition (ICDAR'03), pp 304–309

Walter V (2008) Automatic interpretation of vector databases with a raster-based algorithm. In: The international archives of the photogrammetry, remote sensing and spatial information sciences 37 (Part B2), pp 175–181

Walter V, Luo F (2011) Automatic interpretation of digital maps. ISPRS J Photogrammetry Remote Sens 66(4):519–528

Wuersch M, Egenhofer MJ (2008) Perceptual sketch interpretation. In: Ruas A, Gold C (eds) The 13th international symposium on spatial data handling (SDH 2008). Springer, Montpellier, France, pp 19–38