Lecture Notes
in Geoinformation and Cartography

LNG&C

Fernando Bação
Maribel Yasmina Santos
Marco Painho *Editors*

AGILE 2015

Geographic Information Science as an Enabler of Smarter Cities and Communities

Springer

Lecture Notes in Geoinformation and Cartography

About the Series

The Lecture Notes in Geoinformation and Cartography series provides a contemporary view of current research and development in Geoinformation and Cartography, including GIS and Geographic Information Science. Publications with associated electronic media examine areas of development and current technology. Editors from multiple continents, in association with national and international organizations and societies bring together the most comprehensive forum for Geoinformation and Cartography.

The scope of Lecture Notes in Geoinformation and Cartography spans the range of interdisciplinary topics in a variety of research and application fields. The type of material published traditionally includes:

- proceedings that are peer-reviewed and published in association with a conference;
- post-proceedings consisting of thoroughly revised final papers; and
- research monographs that may be based on individual research projects.

The Lecture Notes in Geoinformation and Cartography series also includes various other publications, including:

- tutorials or collections of lectures for advanced courses;
- contemporary surveys that offer an objective summary of a current topic of interest; and
- emerging areas of research directed at a broad community of practitioners.

More information about this series at http://www.springer.com/series/7418

Fernando Bação · Maribel Yasmina Santos
Marco Painho
Editors

AGILE 2015

Geographic Information Science as an Enabler of Smarter Cities and Communities

Springer

Editors
Fernando Bação
NOVA IMS
Universidade Nova de Lisboa
Lisbon
Portugal

Marco Painho
NOVA IMS
Universidade Nova de Lisboa
Lisbon
Portugal

Maribel Yasmina Santos
School of Engineering
University of Minho
Guimarães
Portugal

ISSN 1863-2246 ISSN 1863-2351 (electronic)
Lecture Notes in Geoinformation and Cartography
ISBN 978-3-319-16786-2 ISBN 978-3-319-16787-9 (eBook)
DOI 10.1007/978-3-319-16787-9

Library of Congress Control Number: 2015936162

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

Since 1998, the Association of Geographic Information Laboratories for Europe (AGILE) promotes academic teaching and research on geographic information at the European level. Its annual conference reflects the variety of topics, disciplines and actors in this research area. It provides a multidisciplinary forum for scientific knowledge production and dissemination and has gradually become the leading Geographic Information Science Conference in Europe.

For the ninth consecutive year, the AGILE conference full papers are published as a book by Springer-Verlag. This year, 48 documents were submitted as full papers, of which 20 were accepted for publication in this volume, after a thorough selection and review process. Thus, we congratulate the authors for the quality of their work and thank them for their contribution to the success of the AGILE conference and book series. We also take this opportunity to acknowledge the contribution of the numerous reviewers for providing us with their thorough judgements. Their work was fundamental to select the very best papers and ultimately for the quality of this volume.

Under the title *Geographic information science as an enabler of smarter cities and communities*, this book tries to envision ways in which GIScience may contribute to a more intelligent and sustainable way of managing resources in our cities and communities.

The scientific papers published in this volume cover a wide range of associated topics. The first part covers challenges in harnessing the power of user-generated content, which will surely change the way we understand and manage urban environments in the future. The second part focuses on methods to extract knowledge and detect changes and anomalies from the massive databases available today, which will keep on getting bigger, providing us with new opportunities to better understand reality. The third part proposes new ways of gaining insight and improving the human mobility conundrum. The fourth, and last part, confronts some foundational problems related to the computational representation and reasoning in GIScience.

Organizing the program of an international conference and editing a volume of scientific papers take time, effort and support. The input from the AGILE Council and Committees was important for us, and we are grateful to all members for their contributions.

We would also like to thank our sponsors, for their kind contributions to the 18th AGILE Conference on Geographic Information Science and Springer-Verlag for their willingness to publish the accepted full papers in their academic series Springer Lecture Notes in Geoinformation and Cartography.

Lisbon — Fernando Bação
Guimarães — Maribel Yasmina Santos
Lisbon — Marco Painho
February 2015

Committees

Programme Committee

Programme Chair Fernando Bação
NOVA IMS, Universidade Nova de Lisboa (Portugal)
Programme Co-Chair Maribel Yasmina Santos
ALGORITMI Research Centre, University of Minho, Guimarães (Portugal)
Programme Co-Chair Marco Painho
NOVA IMS, Universidade Nova de Lisboa (Portugal)

Local Organising Committee

Roberto Henriques, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Adriano Moreira, University of Minho, Guimarães (Portugal)
Jorge Gustavo Rocha, University of Minho, Guimarães (Portugal)
Filipe Meneses, University of Minho, Guimarães (Portugal)
João Moura Pires, FCT-NOVA, Universidade Nova de Lisboa (Portugal)
Armanda Rodrigues, FCT-NOVA, Universidade Nova de Lisboa (Portugal)

Scientific Committee

Adriano Moreira, University of Minho, Guimarães (Portugal)
Alexander Zipf, Heidelberg University (Germany)
Ana Paula Afonso, Faculdade de Ciências da Universidade de Lisboa (Portugal)
Anders Friis-Christensen, European Commission-Joint Research Centre (Italy)
Anne Ruas, IGN (France)
Armanda Rodrigues, FCT-NOVA, Universidade Nova de Lisboa (Portugal)

Bashkim Idrizi, State University of Tetova (Republic of Macedonia)
Beniamino Murgante, University of Basilicata (Italy)
Bin Jiang, University of Gävle (Sweden)
Bisheng Yang, Wuhan University (China)
Bruno Martins, Instituto Superior Técnico (Portugal)
Carlos Granell, European Commission-Joint Research Centre (Italy)
Christoph Schlieder, University of Bamberg (Germany)
Christophe Claramunt, Naval Academy Research Institute (France)
Cidália Fonte, Faculdade de Ciências e Tecnologia,
Universidade de Coimbra (Portugal)
Claus Rinner, Ryerson University (Canada)
Cristina Costa, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Danny Vandenbroucke, KU Leuven (Belgium)
Derek Karssenberg, Utrecht University (The Netherlands)
Didier Josselin, University of Avignon (France)
Dimitris Kotzinos, Université de Cergy-Pontoise (France)
F. Javier Zarazaga-Soria, University of Zaragoza (Spain)
Femke Reitsma, University of Canterbury (New Zealand)
Fernando Bação, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Filipe Meneses, University of Minho, Guimarães (Portugal)
Francesco Pantisano, European Commission-Joint Research Centre (Italy)
Francis Harvey, University of Minnesota (USA)
Francisco J. Lopez-Pellicer, University of Zaragoza (Spain)
Frank Ostermann, European Commission-Joint Research Centre (Italy)
Fred Toppen, Utrecht University (The Netherlands)
Gerard Heuvelink, Wageningen University (The Netherlands)
Gilberto Camara, National Institute for Space Research (Brazil)
Hartwig Hochmair, University of Florida (USA)
Henning Sten Hansen, Aalborg University (Denmark)
Isabel Cruz, University of Illinois (USA)
Itzhak Benenson, Tel Aviv University (Israel)
Jagannath Aryal, Faculty of Science, Engineering & Technology,
University of Tasmania (Australia)
Javier Nogueras-Iso, University of Zaragoza (Spain)
Jérôme Gensel, University of Grenoble (France)
Joaquín Huerta, University Jaume I of Castellón (Spain)
Joep Crompvoets, KU Leuven Public Governance Institute (Belgium)
Jorge Rocha, University of Minho, Guimarães (Portugal)
João Moura Pires, Universidade Nova de Lisboa, FCT-NOVA (Portugal)
Jos van Orshoven, KU Leuven (Belgium)
Karen Kemp, University of Southern California (USA)
Lars Bernard, TU Dresden (Germany)

Lars Harrie, Lund University (Sweden)
Lluis Vicens, Universitat de Girona (Spain)
Luis M. Vilches-Blázquez, Universidad Politécnica de Madrid (Spain)
Maguelonne Teisseire, IRSTEA (France)
Marco Painho, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Maribel Yasmina Santos, University of Minho, Guimarães (Portugal)
Marinos Kavouras, National Technical University of Athens (Greece)
Martin Raubal, ETH Zurich (Switzerland)
Max Craglia, European Commission-Joint Research Centre (Italy)
Michael Gould, University Jaume I of Castellón (Spain)
Michela Bertolotto, University College Dublin (Ireland)
Mike Jackson, University of Nottingham (UK)
Mike Worboys, University of Maine (USA)
Monica Wachowicz, University of New Brunswick (Canada)
Monika Sester, Leibniz University Hannover (Germany)
Nicholas Chrisman, University of Laval (Canada)
Nico Van de Weghe, Ghent University (Belgium)
Oscar Corcho, Universidad Politécnica de Madrid (Spain)
Patrick Laube, University of Zurich (Switzerland)
Pedro Muro Medrano, University of Zaragoza (Spain)
Pedro Cabral, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Peter Atkinson, University of Southampton (UK)
Peter Mooney, National University of Ireland Maynooth (Ireland)
Poulicos Prastacos, Institute of Applied and Computational Mathematics FORTH (Greece)
Ralf Bill, Rostock University (Germany)
Robert Laurini, INSA-Lyon (France)
Robert Weibel, University of Zurich (Switzerland)
Roberto Henriques, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Ross Purves, University of Zurich (Switzerland)
Serena Coetzee, University of Pretoria (South Africa)
Stephan Winter, The University of Melbourne (Australia)
Stephen Hirtle, University of Pittsburgh (USA)
Sven Casteleyn, University Jaume I of Castellón (Spain)
Sven Schade, European Commission-Joint Research Centre (Italy)
Takeshi Shirabe, Royal Institute of Technology (Sweden)
Tapani Sarjakoski, Finnish Geodetic Institute (Finland)
Thomas Blaschke, University of Salzburg (Austria)
Thomas Brinkhoff, Jade University Oldenburg (Germany)
Tiina Sarjakoski, Finnish Geospatial Research Institute (Finland)
Tomi Kauppinen, Aalto University (Finland)
Toshihiro Osaragi, Tokyo Institute of Technology (Japan)

Victor Lobo, NOVA IMS, Universidade Nova de Lisboa (Portugal)
Volker Paelke, Institute of Geomatics–Castelldefels (Spain)
Werner Kuhn, University of California, Santa Barbara (USA)
Wolfgang Reinhardt, Universität der Bundeswehr Munich (Germany)

Contents

Part I
Harnessing the Power of User-Generated Content

Exploring the Potential of Combining Taxi GPS and Flickr Data for Discovering Functional Regions

Jean Damascène Mazimpaka and Sabine Timpf

Abstract The increasing deployment of GPS-enabled devices is leading to an increasing availability of trace data with various applications especially in the urban environment. GPS-equipped taxis have become one of the main approaches of collecting such data. However, we have realized two problems that may limit the effectiveness of use of these taxi GPS data in some applications. Firstly, taxi trajectories represent a very small portion of urban mobility in most of the cities. As a result, without other considerations important information that could come from non-taxi users is excluded. Secondly, advanced applications are built on the analysis of these traces and the context of the movement which is generally obtained from a set of points of interest (POIs). However, considering that POIs are predetermined, we argue that they are a very limited representation of the movement context. The approach we suggest supplements taxi trajectories with crowd-sourced data in an application to discover different functional regions in a city. We cluster the taxi pick-up and drop-off points to discover regions, then semantically enrich these regions using data from Flickr photos and determine the functions of the regions using this semantic information. The evaluation of the approach we performed using large datasets of taxi trajectories and Flickr photos allowed us to identify the potential and limits of the approach which are discussed in this paper.

Keywords Taxi GPS data · Flickr photos · Functional regions · Semantic enrichment · Clustering

J.D. Mazimpaka (✉) · S. Timpf
Department of Geography, University of Augsburg,
Alter Postweg 118, 86159 Augsburg, Germany
e-mail: jean.mazimpaka@geo.uni-augsburg.de

S. Timpf
e-mail: sabine.timpf@geo.uni-augsburg.de

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation
and Cartography, DOI 10.1007/978-3-319-16787-9_1

1 Introduction

The development in Information and Communication Technologies (ICT) has led to an increasing deployment of location aware devices such as GPS-enabled devices. As a result, trace data are increasingly becoming available. GPS-equipped taxis have become a major source of such data enabling a lot of applications ranging from characterising urban space to identifying social dynamics. A detailed survey of the applications of taxi GPS data can be found in Castro et al. (2013).

While the available taxi GPS data may be enough for applications such as determining the average travel time between two hotspots in a city, there are some problems related to the dataset that can affect the effectiveness of some applications. For example, though some studies use taxi trajectories and a set of points of interest (POIs) for discovering the functions of different regions in a city, taxi trajectories represent a very small sample of the urban mobility in most of the cities. Furthermore, POIs are predetermined irrespective of whether they are visited or not. Let us consider an example of an attractive place recommender application based on the movement of tourists. In this application, restaurants may be considered as POIs even if most of the current tourists are vegetarian and hence not interested in non-vegetarian restaurants. While advanced applications of trajectory data analysis take into account the movement context, we argue that the generally predetermined POIs represent a very limited context of the movement.

In addition to the above data related problem, the delineation of activity regions is also a challenging problem. Available work has been attempting to address it by either dividing the study area into a regular grid (Liu et al. 2012), using existing administrative boundaries (Scholz and Lu 2014; Furtado et al. 2013), or subdividing the study area using major roads (Yuan et al. 2012). However, activity regions are irregularly shaped and, as concluded in Rinzivillo et al. (2012), the boundaries of human mobility do not correspond to administrative borders.

Motivated by these observations, we explore an approach for supplementing taxi trajectory data to improve the data representativeness and for using real places of interests which represent the movement context better than general points of interest. We explore the potential and limits of using human mobility in the form of taxi GPS data for delineating regions of interest to people, and using crowd sourced data in the form of Flickr photos to determine the interest of people in these regions as the functions of the regions.

The rest of this paper is organised as follows. In Sect. 2, we present a brief overview of the related work. Section 3 presents the approach we adopted for addressing the problems that we have described while Sect. 4 presents an experimental example of applying the approach. Section 5 discusses the potential and limits of the approach while Sect. 6 concludes and outlines some directions for future work.

2 Related Work

The work presented in this paper lies in the general domain of trajectory data mining. Though this work is related to the general work done in this area such as data mining techniques, this section gives an overview of a few closely related research papers put in two categories: (i) using GPS data for discovering socio-economic functions in a city, and (ii) using crowd-sourced data for place enrichment.

2.1 Discovery of Socio-economic Functions Using GPS Data

Liu et al. (2012) derived urban land use types (commercial, industrial, residential, institutional and recreational) and the dynamics of urban land use changes by mining taxi trajectory data. They explored the temporal variations of both pick-ups and drop-offs in the pixels of the study area and then used the k-means clustering method to classify all pixels in the study area into different classes of traffic source-sink areas. A classification tree method was then used to quantitatively examine the association of traffic source-sink areas with land uses.

Pan et al. (2012) determined the land use types of regions using taxi GPS data. They used a modified version of the DBSCAN algorithm (Iterative DBSCAN) to extract regions based on the characteristics of the data. Four different classification techniques were applied to classify the regions into different social functions (land use types) based on the pick-up and drop-off dynamics.

Yuan et al. (2012) discover the functions of different regions in a city by analysing the human mobility among regions and points of interests (POIs). They infer the functions of each region using a topic-based model which views a region as a document, a region function as a topic, human mobility among regions as words, and a region's categories of POIs as metadata. Their approach partitions the city into regions according to major roads, then for each region, assigns a function distribution vector containing a proportion for each function in the respective region based on the mobility patterns and the POIs.

Our work resembles these studies in analysing the functional distribution within regions and some analysis methods applied such as clustering but the overall framework is different. Furthermore, while these studies partition the whole study area using a regular grid or existing infrastructure (major roads in this case), we attempt to discover the borders of functional regions from the mobility data.

2.2 Place Enrichment Using Crowd-Sourced Data

Brilhante et al. (2013) build a touristic knowledge base from user-generated content and then use the knowledge base for touristic tour recommendation. They extract a

set of points of interest (POIs) from Wikipedia. For each POI, they retrieve the description including a descriptive label, its geographic coordinates, and the set of categories to which the POI belongs. They also retrieve from Flickr the metadata (user ID, timestamp, tags, and geographic coordinates) of photos taken in the same geographic region. The spatial matching of the photos from Flickr and the POIs from Wikipedia produces a rich dataset of places which serves as input to a trip recommender system. While most of the work on semantic enrichment of places relies on the available POI database discarding the places without corresponding POIs in the database, Kisilevich et al. (2010) go a step further to considering also places that have been visited but lack corresponding POIs. In their work on mining travel sequences using geotagged photos from Flickr (Kisilevich et al. 2010), they annotate the place of every geotagged photo in their dataset using the description of the nearby POIs, but also cluster locations of geotagged photos without POIs into places that can further be analysed.

Our work resembles these studies in extracting crowd-sourced data (description of Flickr photos in this case), but the approach of analysing these data is different. While related work analyses each Flickr photo in the context of a place identified by a POI, we analyse the photos in the context of a region and hence we attach more importance to the relation of photos within the same region.

Some other studies are related to ours in delineating regions and using Flickr data but on different problems. Hollenstein and Purves (2010) study the way people describe urban vernacular regions using the tags of Flickr photos. Unlike their study which uses only the tags of Flickr photos, ours integrates two types of dataset (tags of Flickr photo and taxi-based mobility data) to exploit the benefits of each. Though Martins (2011) combines Flickr data with other type of data as we do, he uses a different type of data (land cover data) and his work is only limited to region delineation.

3 Methodology

The approach we propose in this paper is based on the following considerations.

First, the wisdom-of-the-crowds information can supplement taxi trajectory data to improve the representativeness. The taxi trajectories dataset under study may be containing very few trajectories in a particular place which means very few taxis have visited the place and recorded the data. However, there may be other people who have visited the same place and reported some useful information about this place in the form of user-generated content via social media platforms such as Flickr,[1] Twitter[2] or other community contribution based platforms such as

[1] http://www.flickr.com.

[2] http://twitter.com.

Wikipedia.[3] Second, a place that has received some visits in the context of the current application is much more relevant to the application than general points of interest. The information about this place from other sources such as Internet represents much more the movement context for the current application than the attributes of general points of interest.

3.1 Problem Definition

Let s be a point of change of taxi occupancy status. We call it a significant point. Let C be the set of all significant points. A subset C_i of C made of significant points located close to one another is called a cluster and is defined as follows:

$C_i = \left\{ s | \exists t \in C \wedge ||\overrightarrow{st}|| \leq \varepsilon \right\}$ where ε is a fixed Euclidean distance value.

Let R be a set of dense regions occupied by the clusters in C. We call each dense region in R a candidate functional region. That is, a candidate functional region is a polygon estimate of a cluster of significant points: $R_i \approx C_i$.

Let P be a set of Flickr photos of entities found in the candidate functional regions. We call a subset P_i of photos in the candidate functional region R_i the context of R_i. We call each photo in the candidate functional region a contextual entity and it is represented by a pair:

$P_i = \{(a, b) \in F \times G\}$ where F is a set of pre-determined function categories and G is an interval of numbers between 0 and 1; i.e., G = [0, 1]. In short, each photo is represented by the functional category to which it belongs and the corresponding weight. We also define a functional vector V as the vector whose values represent the proportions of specific functions identifying the context of a given region. We call each value in this vector a context value:

$V_i = (x_1, x_2, \ldots, x_n)$, where n = #F and x_j is the context value of the jth function for the ith region.

With the above concepts, the problem of discovering different functional regions in a city using taxi trajectories is translated into determining candidate functional regions, computing their associated functional vectors, and assigning the most likely function to each candidate functional region based on its functional vector.

3.2 Approach

The approach we propose in this study is depicted in Fig. 1. It includes three main steps: region extraction using taxi trajectory data, semantic enrichment of regions

[3] http://en.wikipedia.org/wiki/Main_Page.

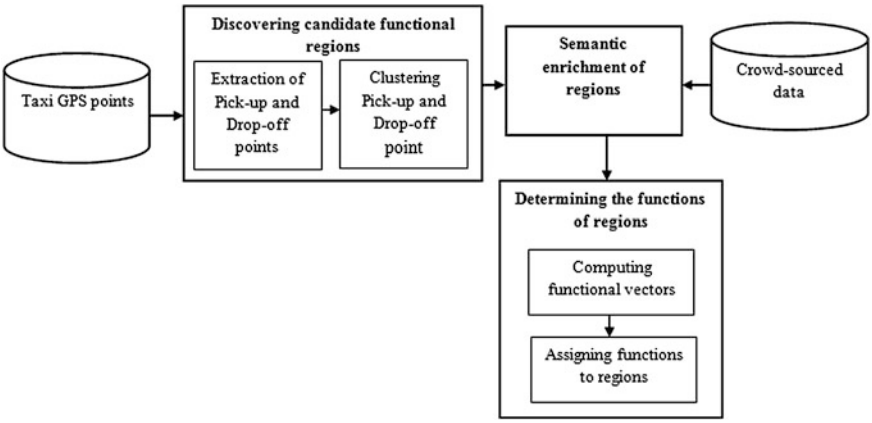


Fig. 1 The approach

using Flickr photos, and computation and assignment of functions to regions using the semantic data collected.

The region extraction starts by extracting the pick-up and drop-off points. Pick-up points are identified as the points of change in occupancy status from *Not occupied* (value 0) to *Occupied* (value 1) while drop-off points are identified as the points of change from *Occupied* to *Not occupied*. Since the exact point of change of occupancy status may not have been recorded, we take the first *Occupied* position after a series of *Non occupied* as the pick-up point whereas the last *Occupied* position before a series of *Non occupied* positions is taken as the drop-off point. The points corresponding to change of occupancy status (pick-up and drop-off points) are then clustered. From this clustering we obtain estimates of the boundaries of functional regions.

The clustering of pick-up and drop-off points is performed using the DBSCAN (Ester et al. 1996), a density based clustering algorithm. This method was chosen because of its advantages with respect to our objective. Firstly, it does not require specifying the number of clusters as input to the clustering process. This matches well with our case where we want to discover functional regions without prior knowledge of how many they are. Secondly, density-based clustering can discover arbitrarily shaped clusters, which is likely to be the case for functional regions.

After discovering candidate functional regions, we semantically enrich these regions. To this end, we download from Flickr public photos taken in the candidate functional regions, recorded in the same time period as our taxi trajectories and labelled with concepts corresponding to the functional categories we want to discover. From each photo, we extracted the title, the description and the tags and

from these we computed the functional weight of the photo. The functional weight of a photo P_i retrieved under the specification of the function category F_x is given by:

$$w_{P_i} = \frac{d}{e} \tag{1}$$

where d is the number of concepts of F_x on P_i and e is the number of all the concepts on P_i. The reasoning behind Eq. (1) is that the more concepts of a given function category a photo has the more likely the photo belongs to this category. The equation penalises a photo that has been tagged with tags that do not correspond to the concepts under consideration. It means that the more unrelated tags a photo has the lesser confident we are that the photo belongs to the category and hence the lower its functional weight should be.

After the semantic enrichment of the regions, we estimate the function of each region using its constituted semantics to turn the candidate regions into functional regions. To this end, we computed a functional vector for each candidate functional region. The functional vector of a candidate functional region represents an estimate of the weights of different function categories in this region. We call the values in the functional vectors "context values". For each function category F_l, the functional vector of the candidate region R_k has a value determined by:

$$v_{kl} = \frac{\sum_{i=1}^{n} w_i}{\sum_{j=1}^{m} w_j} \tag{2}$$

for k = 1,…, #R and l = 1,…, #F and where n is the number of contextual entities associated with the function F_l in a candidate functional region R_k and m is the number of all contextual entities in R_k.

Equation (2) reads: the value at the kth row and lth column of the matrix constituted by the candidate regions as the rows and the function categories as the columns is given by the quotient of the sum of the functional weights of photos associated with the function F_k in R_l to the sum of the functional weights of all photos in R_l. At the final step, we take the function with the highest context value on a given candidate functional region as its function, hence turning the candidate functional region into a functional region. When no function is found to have the highest context value, we consider the region as a mixed function region and take the dominant functions as its function.

Semantic enrichment

In this section we explain in more details our approach of extracting semantic information from Flickr photos for the candidate functional regions discovered. We have pre-defined 5 function categories (residential, commercial, institutional, recreational, and educational) to be discovered. Under each of these function categories, we have identified a number of concepts that are relevant to the category. Following are some of the common concepts that we considered in this work listed depending on their corresponding function category:

Residential: villa, residence, home, apartment
Commercial: shopping mall, supermarket, market, bar, bakery, grocery, restaurant
Educational: university, school, kindergarten, classroom, faculty
Recreational: park, stadium, beach
Institutional: city hall, hospital, health center, municipality, government office

With the identified function categories and corresponding concepts, our aim is to analyse a collection of Flickr photos taken in the geographical extent of a specific candidate functional region in terms of their textual details in order to detect the presence of one or more of the defined concepts on these photos and hence in the corresponding candidate functional region. We have selected only five functional categories and common concepts corresponding to these categories for exploring our approach. However, the approach is designed to work on any number of functional categories and corresponding concepts.

4 Experimental Evaluation

The trajectory dataset used in this experimental evaluation was downloaded from the CRAWDAD[4] website. This dataset contains GPS trajectories of 536 taxis in San Francisco generated in 22 days, from 18 May to 8 June 2008. For each point position in the trajectories, the recorded data are geographic coordinates (latitude and longitude), timestamp, and the taxi occupancy status. The data were recorded at an average sampling rate of 60 s. The whole dataset contains around 11 millions of GPS positions that we stored in Postgres\PostGIS DBMS.

We performed a pre-processing of these data by selecting the data that fall within the study area which is in the San Francisco bay area. We also performed a coordinate system transformation and a data format conversion such as the conversion of timestamp from Unix epoch format to standard date and time format. The restriction of the data to the study area kept 9,845,670 GPS points.

The extraction of pick-up and drop-off points produced 858,614 points including 432,618 pick-up points and 425,996 drop-off points which are distributed within the study area as shown in Fig. 2.

Considering the big size of the dataset which can affect the performance of clustering algorithm and considering that there are repetitive mobility patterns, we selected a subset of the 858,614 GPS points for further processing. The subset was carefully selected to be representative of the overall mobility pattern in the dataset. Since we consider the mobility pattern to repeat itself each week, we selected one week including workdays and weekend days. In order to keep the mobility

[4]Community Resource for Archiving Wireless Data At Dartmouth (http://crawdad.cs.dartmouth.edu).

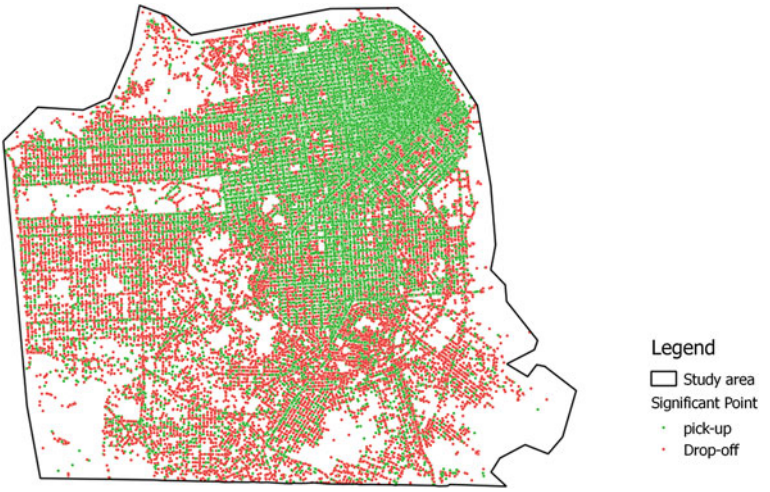


Fig. 2 Spatial distribution of taxi pick-up and drop-off points in the study area

coverage of the whole study area we selected all taxis in operation during the time period considered. We then divided the one week dataset into 14 slices that we consider to represent different mobility characteristics. We considered 7 slices of pick-up and drop-off points recorded in Monday to Friday; between 6.00 and 9.00 (considered to represent the morning rush hour), 9.00–12.00 (morning normal mobility), 12.00–14.00 (midday rush hour), 14.00–16.00 (afternoon normal mobility), 16.00–19.00 (evening rush hour), 19.00–23.00 (evening normal mobility), and 23.00–6.00 (night mobility). We considered another 7 slices of the same time periods for Saturday and Sunday representing the weekend mobility pattern. We clustered the selected sub-dataset using the DBSCAN algorithm implemented in the Weka data mining toolkit with 0.02 and 5 as parameter values for the neighbourhood distance (*Eps*) and the minimum number of points (*MinPts*) respectively. These values were set based on our exploration of the dataset and a number of experiments with different parameter values. The algorithm produced a number of clusters around which we created polygons by building concave hulls. An example of clustering result produced by Weka for the morning rush hour (6.00–9.00) of a workday is shown in Fig. 3. As can be seen from Fig. 3, there is a big cluster, representing the downtown, which needs to be disaggregated. This was achieved through a second level clustering with different parameter values (0.015 and 10 for *Eps* and *MinPts* respectively). The polygons generated were visualised and analysed through the operations of union, intersection, and difference to produce the final candidate functional regions.

Following the discovery of candidate functional regions, we semantically enriched these regions. We downloaded the details including tags and the geographic coordinates of Flickr photos taken in the extent of discovered regions in the time period from 18th May 2008 to 5th December 2014. We considered even the time

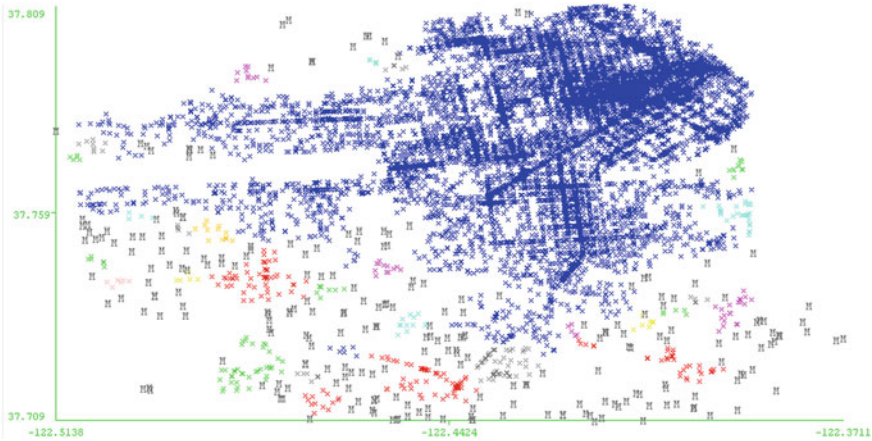


Fig. 3 Clustering result example

period after the period of trajectory data collection to get as many photos as we can and because we assume that the functions of city remain unchanged for a considerable time period. We downloaded the photo details using Flickr Application Programming Interface (API) inside a PHP script that we wrote. The geographic coordinates of the photos enabled us to map the photo locations in the candidate functional regions. Given a candidate functional region and our time period of interest, the script downloaded the details of photos which are labelled with one or more concepts representing our pre-defined function categories. For instance, the script was instructed to download the details of photos tagged with one or more of the keywords *school*, *university*, *kindergarten*, *faculty*, and *classroom* to indicate the *Educational* function category. For each of the extracted photos, we computed the functional weight. The following example illustrates the computation of a functional weight.

A search for photos on the concepts of *Educational* function category returned a photo with the tags *San Francisco*, *University of San Francisco*, *University*. To compute the functional weight of this photo, we use Eq. (1). The value of `d` is the number of tags which are a concept under the considered function (*Educational*) while the value of `e` is the total number of tags on the photo considered. In this example, `d = 2` and `e = 3`. So the functional weight `w` is `2/3 = 0.666`. The functional weight was calculated for each photo in a region.

After the computation of functional weights for all the photos downloaded, we proceeded to the computation of functional vectors of the candidate functional regions using Eq. (2). The procedure is explained with the help of the following example: A region was enriched with 5 photos retrieved under the *Recreational* function having 0.375, 0.041, 0.322, 0.156 and 0.412 respectively as their functional weights. The region has also 2 photos under the *Educational* function with

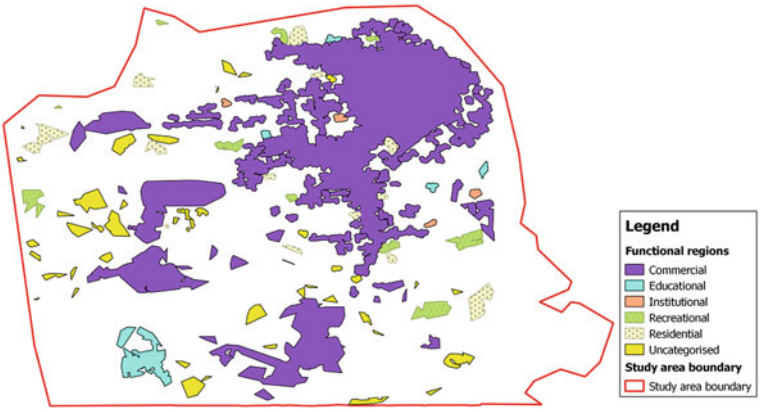


Fig. 4 Functional regions discovered

functional weights of 0.65 and 0.39 respectively. No photos were available for the remaining functions. So the value in the functional vector under *Recreational* is:

`(0.375 + 0.041 + 0.322 + 0.156 + 0.412)/`

`((0.375 + 0.041 + 0.322 + 0.156 + 0.412) + (0.65 + 0.39))` while the value under the *Educational* function is `(0.65 + 0.39)/`

`((0.375 + 0.041 + 0.322 + 0.156 + 0.412) + (0.65 + 0.39))`.

The functional vector becomes `{0, 0, 0.556692, 0.443308, 0}` where the values (called *Context values*) correspond to *Residential, Commercial, Recreational, Educational, and Institutional* respectively. With this functional vector, the region is categorised as a *Recreational* area. The categorisation of candidate functional regions produced functional regions shown in Fig. 4.

The ground truth for evaluating the results obtained in the experiment constitutes of different datasets including the spatial locations of different facilities such as healthcare and recreational facilities downloaded from the central clearinghouse for data published by the City and County of San Francisco (DataSF),[5] the San Francisco land use and land cover dataset downloaded from the USGS[6] website, and the San Francisco Zoning Map downloaded from the Planning Department of the City and County of San Francisco.[7] The Zoning Map data include a detailed land use classification that differentiates for example "Residential-mixed, low density" from "Residential–mixed, moderate density". We have reclassified these data to get classes corresponding to our functional categories. The reclassified land use data are part of the ground truth data shown in Fig. 5. For evaluating our results we overlaid these ground truth data onto the functional regions we discovered and

[5]https://data.sfgov.org/.

[6]U.S. Geological Survey (http://water.usgs.gov/GIS/dsdl/ds240/#poly).

[7]http://www.sf-planning.org/?page=1569.

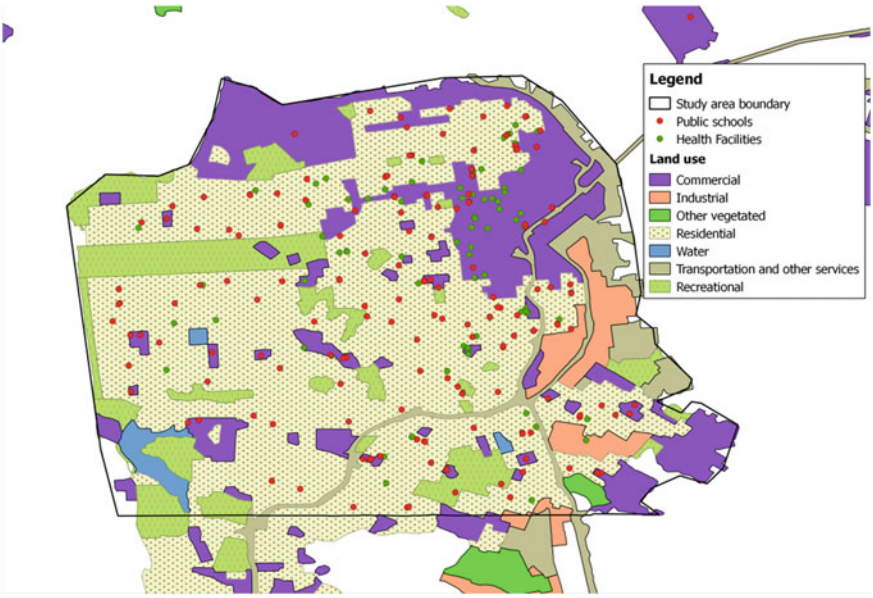


Fig. 5 Ground truth map

visually explored the level of match and mismatch. The overlay of these two datasets is shown in Fig. 6. The results of this evaluation are discussed in the following section.

5 Discussion

The comparison of the results obtained by applying the proposed approach with the ground truth shows that the approach has the potential to discover the locations of different urban functions but with limitations on delineating the functional regions. The potential of the approach lies in the complementary nature of the two types of datasets. Taxi users get in and out in a place because at or near this place there exists some activity that matters for them and the closer these pick-up and drop-off points are the more common is the interest of the users. This common interest is the function at this place and eventually the region. However, this mobility data in itself cannot inform us of what the interest is. On the other hand, crowd-sourced data in general (and Flickr photos in the current case) embed some semantics of the place about which they are collected and cover places where few or no taxis reach, but the locations represented by related data must be densely grouped to build regions.

The limitations of the approach are mainly on delineating regions. This problem is observed differently on different types of functional regions. A look at recreational areas (see Fig. 6) shows that they are discovered as small regions while in

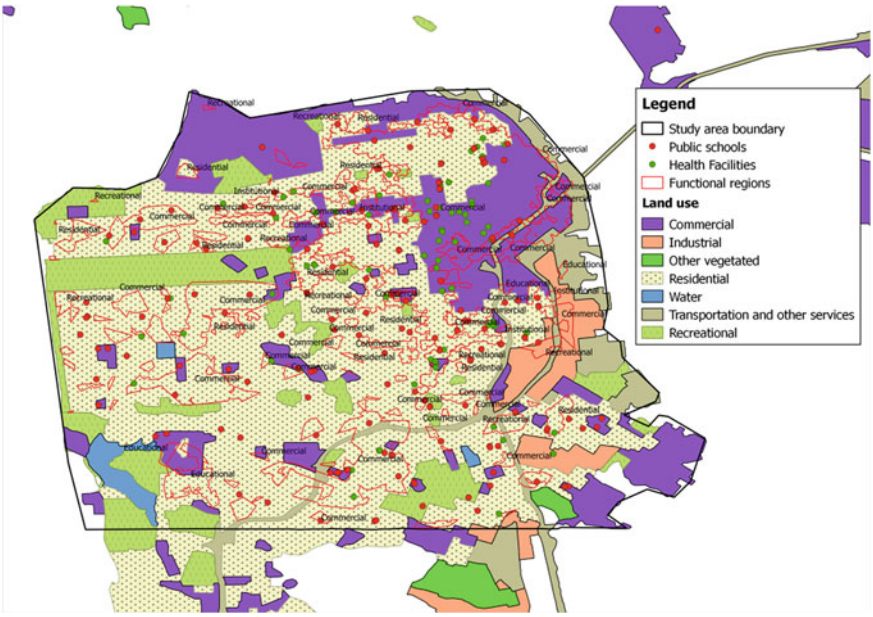
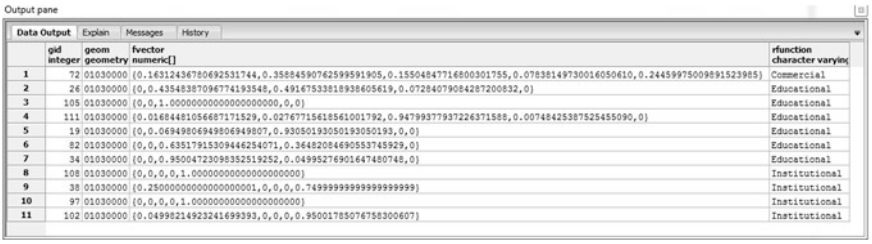


Fig. 6 Discovered functional regions and the ground truth data

reality they are generally wider than many other types. We think that this is due to the way taxis access these areas, i.e., having known entrances where they drop and pick their users without getting inside the area. This is the case of the "Golden Gate Park" which is a very wide recreational area in the north west of the study area but which has not been delineated (see Fig. 6). Instead, some small recreational areas have been discovered near the corners of the park which might correspond to the main entrances of the park to justify our opinion. While the residential area expands over a wide area, it is generally discovered as small isolated regions. In our opinion, this may be due to sparse taxi GPS data in these areas. Our opinion can be justified by observing the location of these areas which is mainly the central and western part (see Fig. 5) and the corresponding western part of the clustering results on Fig. 3. The clustering may have left the large part of these sparse data unclustered.

In the high activity areas such as the downtown the mobility is also high, leading to a dense wide region in which it is difficult to identify individual functional sub-regions. This can be seen from the example of clustering result shown in Fig. 3 where the downtown was discovered as one large dense cluster. In an effort to extract different regions in this area, we applied a multilevel clustering on it which led to the discovery of smaller different regions as seen in Fig. 4. However, the discovery of different functional regions in hotspot areas such as the downtown is still challenging. We note a trade-off between setting suitable clustering parameters and avoiding very tiny regions hardly seen on the map. Furthermore, the challenge in discovering functional regions in such areas is seen on the mixed nature of its


Output pane

Data Output | Explain | Messages | History

	gid integer	geom geometry	fvector numeric[]	rfunction character varying
1	72	01030000	{0.16312436780692531744,0.35884590762599591905,0.15504847716800301755,0.07838149730016050610,0.24459975009891523985}	Commercial
2	26	01030000	{0,0.43548387096774193548,0.49167533818938605619,0.07284079084287200832,0}	Educational
3	105	01030000	{0,0,1.0000000000000000000,0,0}	Educational
4	111	01030000	{0.01684481056687171529,0.02767715618561001792,0.94799377937226371588,0.00748425387525455090,0}	Educational
5	19	01030000	{0,0.06949806949806949807,0.93050193050193050193,0,0}	Educational
6	82	01030000	{0,0,0.63517915309446254071,0.36482084690553745929,0}	Educational
7	34	01030000	{0,0,0.95004723098352519252,0.04995276901647480748,0}	Educational
8	108	01030000	{0,0,0,0,1.0000000000000000000}	Institutional
9	38	01030000	{0.25000000000000000001,0,0,0,0.74999999999999999999}	Institutional
10	97	01030000	{0,0,0,0,1.0000000000000000000}	Institutional
11	102	01030000	{0.04998214923241699393,0,0,0,0.95001785076758300607}	Institutional

Fig. 7 Sample functional vectors

functions. For instance, for most of the regions we could easily find an outstanding function as the function of the region even with some other functions being not represented in these regions. Conversely, all the functions are represented in the wide region corresponding to the downtown and even the *Commercial* function assigned to it dominates with a small proportion and a small difference to most of the other functions as shown in Fig. 7 by the functional vector of this region. The functional vector is made of 5 values corresponding to *Residential*, *Commercial*, *Educational*, *Recreational*, and *Institutional* respectively. As it can be seen in Fig. 7 where the record of the downtown area is the first (with *gid* = *72*), the *Commercial* function dominates with a value of 0.3588459 which is small compared to the maximum value of 1 and with a small difference to *Institutional* (0.2445997), *Residential* (0.1631243) and *Educational* (0.1550484). We assume that this observation will become even stronger if more functional categories are considered. Another observation from the functional vector of the downtown region is that the *Recreational* function has a much smaller value (0.0783814) as indeed we expected.

The major limitation observed on the characterisation of regions is that some areas can be discovered as candidate functional regions but remain uncategorized. A close analysis of these regions shows that they have generally a small area. In Fig. 6, on a total of 123 candidate functional regions there are 34 uncategorised regions of which 30 are very small. We think that these are regions which have not attracted the interest of Flickr photo takers possibly because of a limited number of attractive objects. In order to validate our assumption, we observed the detailed land use classification of the western part of the map where most of these uncategorised regions are located. We found that they are located on areas of which the land use class is either “Residential–house, one family” or “Residential–house, two family” which are likely not attractive to photo takers. This is a general problem that has been observed also in some previous work (Rattenbury et al. 2007): crowd-sourced data in general are unevenly distributed such that a church dome may have hundreds of Flickr photos while in a considerable neighbourhood of it no other object has even one single photo.

While the accuracy of region delineation is not so good, the accuracy of region characterisation (inferring functional categories from Flickr tags) is high because only few and very small regions were not characterised and for those characterised

the characterisation matches the ground truth. Considering the few functional categories and very common corresponding concepts, we have limited the accuracy assessment of our discovery approach to the observation of the overlay of the discovered regions and the ground truth data. However, a future work with more functional categories and corresponding concepts will also perform a detailed accuracy assessment using methods such as confusion matrices.

6 Conclusions and Future Work

In this paper we proposed an approach for supplementing taxi GPS data with crowd-sourced data for region discovery. The problem addressed was instantiated on a case of Flickr photos as the crowd-sourced data for discovering regions of different functions in a city. Various applications can benefit from the discovery of functional regions in a city. Urban planners can use the discovered information for updating the city land use map, the localisation of different activities and the deployment of resources. This information can support people in general in recommending and in choosing location for businesses, advertisement and tourism. The approach we proposed performs a clustering of the taxi pick-up and drop-off points to determine regions and then uses data extracted from Flickr photos to enrich these regions semantically. The analysis of this semantic information computes a functional vector for each region which is finally used to categorise the region in one function from a pre-defined set of function categories. We evaluated the proposed approach and elaborated on its potential and limits.

The novelty of our approach compared to related work is the integration of two different types of data to exploit the benefits of each. Taxi GPS data have a high potential to reveal also regions of important functions that might not be revealed by Flickr photos because they are generally less attractive to photo takers. An example of such regions is residential areas. On the other hand, Flickr photos carry textual descriptions potential for inferring the function of the place where they were taken. Another advantage of our approach is that it can also enable a fuzzy classification of discovered regions into functional categories. For each categorised region, the functional vector contains estimates of shares of different functional categories in it. Instead of taking the dominant function, it is possible to take the first two or three functions as the mixed functions of the region especially in case the functions have almost equal values.

In view of the identified limitations of the proposed approach, the future work includes the exploration of more advanced approaches for discovering and delineating regions. Our initial idea in this line is to partition the study area, based on the density of the taxi GPS points and choose suitable clustering parameters for each partition. We think that this localised clustering can handle sparse data areas and wide dense areas such as the downtown better than the global clustering commonly applied. In the line of semantic enrichment of regions, we will extend our approach of extracting the semantics of regions by considering additional functional

categories and exploring more advanced techniques of processing concepts under different functional categories. Another line of work is on integrating other sources of information to tackle the lack of information for some areas.

References

Brilhante, I., Macedo, J. A., Nardini, F. M., Perego, R., & Renso, C. (2013). Where shall we go today? Planning touristic tours with tripbuilder. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM'13)* (pp. 757–762). New York: ACM.

Castro, P. S., Zhang, D., Chen, C., Li, S., & Pan, G. (2013). From taxi GPS traces to social and community dynamics: A survey. *ACM Computing Surveys, 46*(2), 1–17.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)* (pp. 226–231).

Furtado, A. S., Fileto, R., & Renso, C. (2013). Assessing the attractiveness of places with movement data. *Journal of Information and Data Management, 4*(2), 124–133.

Hollenstein, L., & Purves, R. (2010). Exploring place through user-generated content: Using Flickr to describe city cores. *Journal of Spatial Information Science, 1*(1), 21–48.

Kisilevich, S., Keim, D., & Rokach, L. (2010). A novel approach to mining travel sequences using collections of geotagged photos. In M. Painho, M. Y. Santos, H. Pundt, W. Cartwright, G. Gartner, L. Meng, & M. P. Peterson (Eds.), *Geospatial Thinking* (pp. 163–182). Lecture Notes in Geoinformation and Cartography. Berlin: Springer.

Liu, Y., Wang, F., Xiao, Y., & Gao, S. (2012). Urban land uses and traffic 'source-sink areas': Evidence from GPS-enabled taxi data in Shanghai. *Landscape Urban Planning, 106*(1), 73–87.

Martins, B. (2011). Delimiting imprecise regions with georeferenced photos and land coverage data. In K. Tanaka, P. Fröhlich, & K. S. Kim (Eds.), *Web and wireless geographical information systems* (pp. 219–229). Berlin: Springer.

Pan, G., Qi, G., Wu, Z., Zhang, D., & Li, S. (2012). Land-use classification using taxi GPS traces. *IEEE Transactions on Intelligent Transportation Systems, 14*, 113–123.

Rattenbury, T., Good, N., & Naaman, M. (2007). Towards automatic extraction of event and place semantics from Flickr tags. In *Proceedings of the Thirtieth International ACM SIGIR Conference, (SIGIR 07).*

Rinzivillo, S., Mainardi, S., Pezzoni, F., Coscia, M., Pedreschi, D., & Giannotti, F. (2012). Discovering the geographical borders of human mobility. *Künstliche Intelligenz, 26*, 253–260.

Scholz, R. W., & Lu, Y. (2014). Detection of dynamic activity patterns at a collective level from large-volume trajectory data. *International Journal of Geographical Information Science, 28* (5), 946–963.

Yuan, J., Zheng, Y., & Xie, X. (2012). Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)* (pp. 186–194).

A Semantic Region Growing Algorithm: Extraction of Urban Settings

Heidelinde Hobel, Amin Abdalla, Paolo Fogliaroni and Andrew U. Frank

Abstract Recent years have witnessed a growing production of Volunteer Geographic Information (VGI). This led to the general availability of semantically rich datasets, allowing for novel ways to understand, analyze or generalize urban areas. This paper presents an approach that exploits this semantic richness to extract *urban settings*, i.e., conceptually-uniform geographic areas with respect to certain activities. We argue that *urban settings* are a more accurate way of generalizing cities, since it more closely models human sense-making of urban spaces. To this end, we formalized and implemented a *semantic region growing* algorithm—a modification of a standard image segmentation procedure. To evaluate our approach, shopping areas of two European capital cities (Vienna and London) were extracted from an OpenStreetMap dataset. Finally, we explored the use of our approach to search for urban settings (e.g., shopping areas) in one city, that are similar to a setting in another.

Keywords Semantic region growing · Image segmentation · Urban settings · Place affordances

H. Hobel (✉)
Doctoral College Environmental Informatics, SBA Research, Vienna University of Technology, Wien, Austria
e-mail: hobel@geoinfo.tuwien.ac.at

A. Abdalla · P. Fogliaroni · A.U. Frank
Department for Geodesy and Geoinformation, Vienna University of Technology, Wien, Austria
e-mail: abdalla@geoinfo.tuwien.ac.at

P. Fogliaroni
e-mail: fogliaroni@geoinfo.tuwien.ac.at

A.U. Frank
e-mail: frank@geoinfo.tuwien.ac.at

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_2

1 Introduction

Human conceptualization of space is one of the main research questions in Geographic Information Science, Spatial Information Theory, and Urban Planning and many other disciplines (Lynch 1960; Mark and Frank 1991; Tuan 1979). Many have studied the way humans navigate through or reason about space (Lynch 1960; Raubal 2001). Building upon the findings of such studies, computational models and applications have been developed that simulate human conceptualization in order to improve usability of software or to equip computer systems with basic intelligence.

A particularly interesting question concerns the conceptualization of places: The ambiguous meaning of the term poses a considerable challenge to knowledge engineers whose task is to design computational models of places. As of today, the most commonly adopted strategy is to represent places by means of points of interest (POIs). This approach, however, disregards many of the aspects that seem to characterize human conceptualization of places: (i) there is empirical evidence (Montello et al. 2003) that people typically conceive a place as a region; (ii) different persons tend to associate different spatial footprints to the same place (Montello et al. 2003); (iii) there are indications (Schatzki 1991, p. 655) that conceptualization of a place relies on the activities that are possible to carry out at that spatial location—i.e., what some refer to as place affordances (Jordan et al. 1998). Accordingly, the approach of representing places with POIs suffers from several drawbacks: places are indicated as specific points rather than vague or approximated regions; while a POI is associated with a precise feature type, the place affordances are not explicitly indicated and it is up to the user to map from an activity (e.g., to eat) to a feature type (e.g., a restaurant or a fast-food). Going even further and focusing our attention on activities, it is easy to see that activities are usually not restricted to a single place and have an extent in space and time that involves several places of different kinds. Shopping, for example, can involve sitting down at a cafe, or going to a bank to withdraw money. Humans are able to search for areas that *afford* an activity without having to specify the exact type of place they are looking for. For example, if the task is "to buy a pair of shoes and perhaps a coat", humans can, based on experience or knowledge, think of areas where they are most likely to find such things (e.g., a shopping street or shopping mall). In such a case, the individual shop is less of concern since the exact object to buy is not determined yet. Rather, it is the constellation or setting of shops and maybe restaurants, that is of importance when attempting to find an area suitable for an activity.

Inspired by techniques employed in image processing and land use detection, we present a semantic region growing algorithm that exploits tag information from OpenStreetMap[1] data to produce areas corresponding to a setting of interest. The

[1] http://www.openstreetmap.org.

question of how to find such settings is, to our knowledge, not well addressed and this paper presents preliminary results of an attempt to extract urban settings based on activities (or affordances). The underlying hypothesis is that people form regions by mentally grouping space into conceptually homogeneous areas in terms of the activities they potentially offer. Therefore, place types (represented by tags) are employed as a means of computing potential activities. This work aims at extending an ongoing effort to find generalization techniques of urban areas that transcend common administrative partitions. The contributions of this paper are twofold:

- An implementation of a semantic region growing algorithm, that can be used to find Urban Settings from point data with place-type information (POI's);
- A discussion and preliminary evaluation of using the approach to search for similar areas in other cities.

The paper is structured as follows: Sect. 2 discusses some of the literature on Place and Settings, introduces some work on Image Processing, and outlines OpenStreetMap's knowledge representation scheme and main data quality issues. Section 3 introduces the proposed method to find appropriate settings. Section 4 presents preliminary results of a case study and Sect. 5 will discuss the outcomes, limitations and future work. In Sect. 6, we conclude our work.

2 Related Work

In this section, we investigate related work concerning places and settings, image segmentation, and OpenStreetMap.

2.1 Places and Settings

The concept of *place* plays an increasingly important role in GIScience (Winter et al. 2009; Winter and Truelove 2013) and the ontological discussion about how to model it is ongoing (Couclelis 1992; Humayun and Schwering 2012; Jones et al. 2001; Vasardani et al. 2013; Winter and Truelove 2013). Many suggest that the semantics of the term *Place* is tightly bound to the idea of affordance and activities (Jordan et al. 1998; Scheider and Janowicz 2014). As a matter of fact, drawing the connection of action to place, is essential for the ability to plan (Abdalla and Frank 2012). Schatzki asserted that: "[…] places are defined by reference to human activity" (Schatzki 1991, p. 655). He positions human activities as the central concept for understanding the construction of places. Furthermore, he explains that such representations of places organize into settings, local areas and regions. This general notion of hierarchical structuring of space is relatively undisputed and

supported by findings of other researchers (Couclelis and Gale 1986; Freundschuh and Egenhofer 1997; Montello 1993; Richter et al. 2013). How these levels of abstractions are formed, though, is unclear. For example, common administrative units of abstraction do not always correspond to what people have in mind about regions (Meegan and Mitchell 2001).

The focus of this work lies on *settings* which, according to Schatzki, can either be demarcated by barriers (e.g., apartment building) or identified by bundles of activities that occur in them (e.g., a park, or shopping street). Ontologically speaking, they can either be categorized as entities of *bona fide* (i.e. physical, sharp, crisp) or *fiat* (i.e. non-physical, imaginary, human-driven) type (Smith 1995). Since this work is concerned with entities larger than apartment buildings, such as shopping areas, fiat objects will be the main type of inquiry. The entities are therefore of the vista-space scale (Montello 1993), since they can be learned by human activity.

2.2 Image Segmentation

Image segmentation builds on the idea of grouping pixels into areas. Professionals in Remote Sensing make use of image segmentation techniques to categorize satellite images in terms of land use or land cover, e.g., see (Shimabukuro et al. 1998). One implementation of such an image segmentation algorithm is known as *Region Growing*, where homogeneous pixels of the image are coalesced (Adams and Bischof 1994; Fan et al. 2005). Starting from a seed pixel, the algorithm recursively expands into the adjacent neighborhood and classifies each pixel in it as similar or not, according to certain constraints. All adjacent pixels similar to the initial pixel are then merged into a group, referred to as *segment*.

2.3 OpenStreetMap

OpenStreetMap is a web project, whose main goal is to create a digital map of the entire world, and is essentially the prototype of Volunteer Geographic Information (Goodchild 2007). The geometric footprint of spatial features is represented by means of a simple and exceptionally flexible scheme consisting of

- *nodes*: pairs of coordinates (longitude and latitude) used to represent point features;
- *ways*: lists of nodes used to represent line and surface features;
- *relations*: sets of nodes, ways, or other relations mainly used to represent features consisting of several parts.

The thematic or semantic aspect of spatial features is managed through a tagging system where each geometric feature is described by an arbitrary number of tags. As the OpenStreetMap project evolved and prospered over time, its community developed a set of *tagging guidelines* that describe which tags should be used for a specific feature. Before contributing new information to the database, mappers are asked to carefully read these guidelines. Yet, they are neither obligated to respect such guidelines nor are their contributions subject to rigorous control.

It has been shown that geometric-wise the OpenStreetMap dataset is rapidly approaching the coverage and the precision of commercial ones (Zielstra and Zipf 2010). The freedom granted by the tagging system yielded a semantically very heterogeneous dataset (Mooney and Corcoran 2012). Thus, different volunteers tag the same feature differently or, conversely, use the same tag to annotate conceptually different features. Moreover, some recent works (D'Antonio et al. 2014; Keßler and de Groot 2013) investigated the possibility of assessing the trustworthiness of VGI data by analyzing the historical evolution of features in a dataset. However, semantic quality of VGI data remains, at the time of writing, a major issue.

3 Conceptual Spatial Region Growing Algorithm

In this section, we explain the idea of conceptual settings, the formalization of setting segmentation, and the implementation of the algorithm.

3.1 Conceptual Spatial Settings

City maps are cartographic representations of spatial data partitioning space into discrete chunks that represent physical or social (administrative) objects. These objects are either defined by their physical extent or by authoritative institutions. Following the argument of Schatzki (1991), there are places that are falling into the same abstract category due to certain constellations of things present and activities possible. It is the focus of this work to use a data driven approach to find such conceptual spatial settings derived from the places an activity needs. For instance, the description of a shopping area—exemplarily illustrated in Fig. 1—should obviously contain shops, but can also include parking spots, playgrounds, restaurants, and so on.

Furthermore, once homogeneous areas have been defined, a formal description of the area offers abilities to search, compare or cluster such regions. Figure 1 depicts three conceptual shopping settings together with their respective frequency

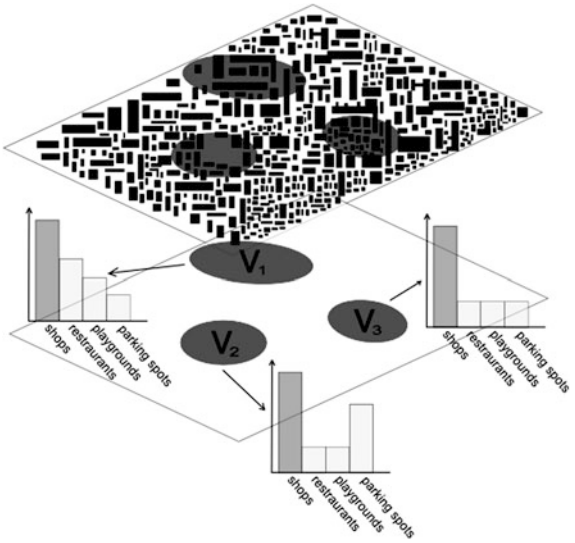


Fig. 1 Schematic visualization of conceptual shopping areas: V_1, V_2, V_3

distribution tags. While each one of them contains shops, places like parking spots and restaurants are also part of the constellation "shopping area" (see Fig. 1).

Fine-grained or significant differences in place constellations can reveal how much the composition of a setting is suitable for someone's preferences, or can be used to identify flaws in the naturally evolved or planned structure of a city. For instance, when people are required to drive with a car due to an inefficient public transport system, or out of personal necessity, shopping areas with parking spots are certainly more attractive destinations. Cities without dedicated parking spots in the vicinity of shopping areas will ideally have an efficient public transport. Therefore, using an aggregate description of the coalesced areas as a semantic signature enables comparison and assessment of conceptual settings. The composed area offers not only single place affordances, but rather encompasses a set of affordances which are seamlessly interconnected.

3.2 Formalization

The goal of the proposed approach is to identify areas according to the activities *afforded* by constellation of places contained in it. We draw inspiration from a technique used in image segmentation and adapted the region growing algorithm (Adams and Bischof 1994) to become a *semantic region growing* algorithm in the following manner:

1. The area of interest $\mathscr{M}$ (a city map in our case) is partitioned into n non-overlapping cells $C = \{c_i : i = 1, \ldots, n\}$ such that $\mathscr{M} = \bigcup_{i=1}^{n} c_i$.
2. The essential concept of our region growing approach is that of a description D: a formula consisting of one or more predicates specifying the membership of a single cell c_i to a specified setting S, e.g., a description can be: "contains at least one shop and restaurant".
3. What in image segmentation jargon is called a *segment*, is directly comparable to a setting: a set of contiguous cells satisfying the same description D. A setting $S \subseteq C$ is a subset of the cell partition C and is called *complete* if it cannot be extended further with adjacent cells.
4. The segmentation of a map $\mathscr{M}$ according to a description D produces a (possibly empty) set $\mathscr{S}$ of settings such that $\bigcup_{S \in \mathscr{S}} S \subseteq C$. A segmentation $\mathscr{S} = \{S\}$ is called *complete* if it consists of only one setting such that $S = C$.
5. As image segmentation relies on a similarity function that is used to decide if two neighbor pixels are similar, so does our approach rely on a Boolean function f_{sim} which, given a cell c and a description D, verifies whether c adheres to D.
6. Settings identified through the same description D are pairwise disjoint, i.e. it holds that for all $x, y \wedge x \neq y : S_x \cap S_y = \emptyset$. Settings that adhere to different descriptions can overlap, e.g., a park that crosses a shopping street.

3.3 Implementation

Semantic region growing as used here, is aimed at segmenting or extracting settings according to a description D and a set of m cells, referred to as *seeding cells* $C_{seed} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_m\}$. In the case that a seeding cell $\tilde{c} \in C_{seed}$ matches a given description D—i.e., $f_{sim}(\tilde{c}, D) = TRUE$—and it is not yet classified as a member of another setting adhering to the same description D, $\tilde{c}$ will be the starting point of a new setting: A recursive process extends the starting cell until the adjacent neighborhood does not adhere any longer to the description D. We can either process all cells as seeding cells ($C_{seed} = C$), or find all cells in C that adhere to the description D and use them as C_{seed}—both cases yield a robust result in contrast to random seed generation. For instance, if C_{seed} contains only five seeding cells, then the result will be at most five segments/settings. Note that a settings S will not be identified by the algorithm if $S \cap C_{seed} = \emptyset$—i.e., if no seeding cell lies within S. Additionally, it is possible that during the growing process starting from a seeding cell $\tilde{c}_i$ and building a setting S_i, another seeding cell $\tilde{c}_j$ is integrated in S_i. When the algorithm will process the seeding cell $\tilde{c}_j$, this will not give raise to a new setting since it has already been assigned to the setting S_i. The *semantic region growing* technique is implemented as shown in Algorithm 1.

Algorithm 1 Semantic Region Growing Algorithm

```
1:  procedure FINDSETTINGS( description D, seeding cells C_seed, cells C)
2:      new 𝒮 ← ∅
3:      for all c̃ ∈ C_seed do
4:          if (∀S_i ∈ 𝒮 : c̃ ∉ S_i) ∧ f_sim(c̃, D) then
5:              new S ← {c̃}
6:              𝒮 ← 𝒮 ∪ {S}
7:              REGIONGROWING(D, c̃, C, S, 𝒮)
8:          end if
9:      end for
10: end procedure

11: procedure REGIONGROWING( description D, cell c, setting S, list of settings 𝒮, cells C )
12:     new N ← neighbours(c, C)
13:     for all n ∈ N do
14:         if (∀S_i ∈ 𝒮 : n ∉ S_i) ∧ f_sim(n, D) then
15:             S ← S ∪ {n}
16:             REGIONGROWING(D, n, S, 𝒮, C)
17:         end if
18:     end for
19: end procedure
```

As can be seen in the implementation of Algorithm 1, the size of the adjacent neighborhood of a cell can be adapted by using a customized implementation of $neighbours(c, C)$ to specify requirements such as larger or restricted neighborhoods. In any case, a larger neighborhood can be used to ensure a better coverage or restrictions can be used to separate settings.

4 Case Study

In this section, we present a first evaluation of our approach, explain in an illustrative use case how to differentiate between settings of the same conceptualization, and analyze the results.

4.1 Setup

For a first evaluation of our approach, we attempted to identify shopping areas in two cities. Therefore, we collected data from Metro Extracts,[2] a website that provides parts of the OSM datasets for cities and their surroundings. We

[2] http://metro.teczno.com/.

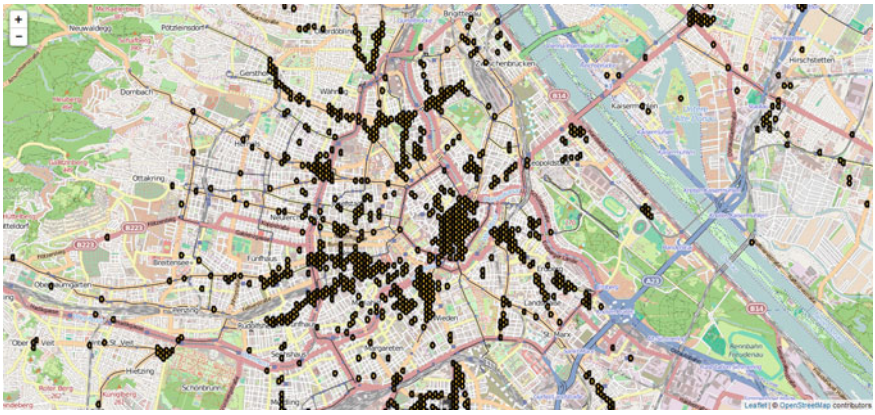


Fig. 2 Visualization of the results identified by the semantic region growing algorithm in Vienna

downloaded and parsed the datasets for Vienna and London. By using GeoTools,[3] an open source library for geospatial data, we set up a fine grained hexagonal grid, whereby the side length of the cells was set to 0.0005 degrees, and preprocessed the OSM data by assigning the nodes and their tags to the cells. The rules for our description are based on the following assumptions.

> A cell has to encompass at least two places where you can *shop* (i.e., shops of every type) or a cell has to encompass at least two tags that relate to places where someone can get something to eat or drink a coffee (e.g., restaurants, fast-foods, cafes).

These simple constraints were sufficient to find the commonly-known *shopping areas* in Vienna, plus many smaller clusters that can be interpreted as local shopping and leisure areas. The segmentation result for Vienna is shown in Fig. 2. Using the same description, we employed the algorithm on the dataset of London and obtained a comparable result (see Fig. 3).

Arguably, there is no *hard* method to evaluate the result, since the topics of interest are conceptual settings, that do not really allow for a ground truth. Nevertheless, an estimation of feasibility is still possible, either by looking at descriptions found on the internet (e.g., Wikipedia, Tourism Guides) or by comparing the results to expert knowledge (i.e., people familiar with the city). Indeed, Mariahilferstraße and Oxford Street are well-known shopping streets that have been correctly identified as part of shopping settings by our algorithm. Also, detailed explorations of some other clusters identified in the Vienna dataset, consistently revealed that all larger found regions can be considered shopping areas.

[3] http://docs.geotools.org/.

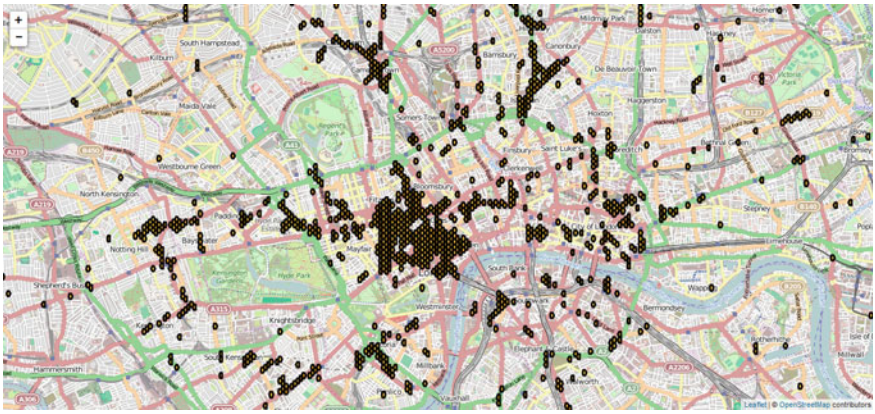


Fig. 3 Visualization of the results identified by the semantic region growing algorithm in London

4.2 Use Case Example and Analysis

Now that the areas are identified, the next step is to compare the identified shopping areas in Vienna and London. Consider the following scenario.

> Alice grew up in London and she knows from experience that in the *urban setting* of Oxford Street there are plenty of places to withdraw money (i.e., ATMs and banks), that there is a big choice of cafes and restaurants to go for lunch or get something to drink. Also, there is a large diversity of shops and several tourism attractions that she sees when moving from a shop to another. Alice plans a trip to Vienna and she would like to find, in advance, areas of the city that are similar to her idea of Oxford Street.

To model these preferences and action possibilities, we defined the following four features, which will later be used to define a similarity-distance to other identified shopping areas/settings:

1. The number of tags in a setting of type bank or ATM n_1
2. The number of tags in a setting of type restaurant or cafe or fast food n_2
3. The number of tags in a setting of type tourism n_3
4. The number of different shopping types (i.e., subcategories of shops) n_4

We denote by $\tau_{(S)}$ the total number of cells in a given segment/setting S. We set the absolute values n_1, n_2, n_3 and n_4, which we defined in the list above, in relation to the area of the setting, which yields normalized density values (where m is the number of defined features):

$$r_i = \frac{n_i}{\tau_{(S)}} \quad \forall i = 1, \ldots, m \tag{1}$$

To explore the similarity in respect to our defined feature vector, we are now considering the following distance measure:

$$\sum_{i=1}^{m} |r_i^{(S_1)} - r_i^{(S_2)}| \tag{2}$$

Equation (2) formalizes the sum of the absolute values of the differences between corresponding features for two settings with normalized values $r_{(\cdot)}^{(S_1)}$ and $r_{(\cdot)}^{(S_2)}$.

Based on the use case scenario explained above, Alice wants to know what are similar shopping areas in comparison to London's Oxford Street in Vienna. Therefore, we denote by $r_i^{(S_1)}$ the values of Oxford Street and make a comparison with the larger sized extracted conceptual shopping settings of Vienna, since we normalized the data based on the size of the settings. According to the total deviation [see Eq. (2)] the best matching setting is the area found around the *Inner City* and the second one is the cluster around the lower part of *Mariahilferstraße*, which is illustrated in Fig. 4.

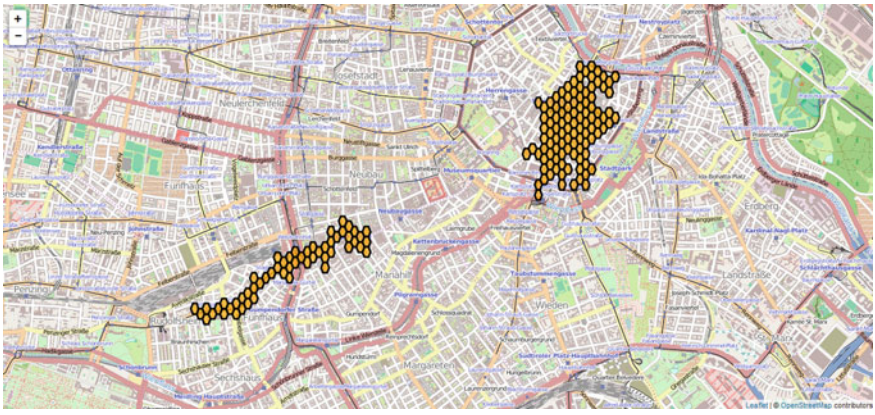


Fig. 4 Visualization of the identified shopping areas in Vienna, which are most *similar* to the Oxford Street (London)

Figure 5 illustrates the deviations of the defined preferences between the areas of *Inner City* and *Oxford Street* (black), as well as the *Mariahilferstraße* and *Oxford Street* (grey). The total deviation, which is defined through the similarity-distance given in Eq. (2), can be read off the *absolute deviation* axis. Thereby, a lower deviation is indicated when the instance in comparison, i.e. the line for *Inner City* or *Mariahilferstraße*, is nearer to the center. In this case, it can be clearly seen that the total deviation of the Inner City is lower than the deviation of the *Mariahilferstraße* in comparison to the *Oxford Street* area. To enable a more fine grained comparison, we plotted for each $i = 1,\ldots, 4$ the value of $|r_i^{(S_1)} - r_i^{(S_j)}|$, which is the single deviation on an independent axis. In the previous formula, the variable j stands for either 2 or 3, which corresponds to *Inner City* or *Mariahilferstraße*, respectively. We briefly elaborate on the individual feature differences according to Fig. 5:

1. Regarding the density of banks and ATMs, the area found in the *Inner City* as well as the one around *Mariahilferstraße* are both relatively close to the area that contains *Oxford Street*.
2. In terms of the density of tourism attractions *Mariahilferstraße* is a bit closer to *Oxford Street* than the *Inner City*.
3. The higher deviations in terms of density of Restaurants, Cafe, and Fast Food places, and shop diversity of the *Mariahilferstraße* indicate that the *Inner City* is more similar to the *Oxford Street*.

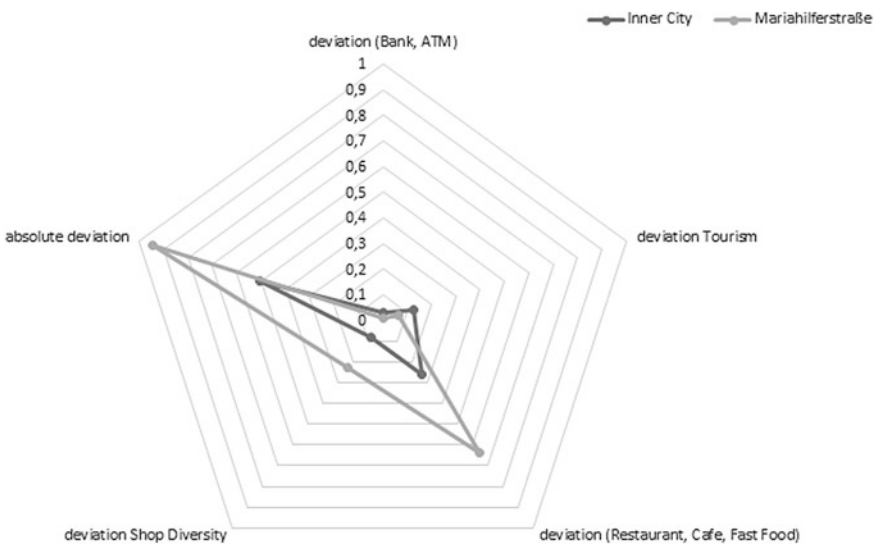


Fig. 5 The deviations of the conceptual shopping area Oxford Street (London) to the conceptual Mariahilferstraße and Inner City (Vienna) in respect to the defined feature vector

This section showed that our approach allows to search for urban settings. By discriminating areas suitable for certain activities, it is possible to compare those on the basis of a formal description of preferences. The type descriptions of places, as available in OpenStreetMap, do provide the basis for a mapping of activity possibilities to conceptual Settings.

5 Discussion, Limitations and Future Research Directions

The work presented in this paper is an excerpt of ongoing work that investigates a data driven approach of urban area generalization. Aside from many other challenges, this research has to face, two are most striking: (1) The choice of the description criteria D and evaluation of the results; and (2) the semantic ambiguity as well as incompleteness of VGI data.

The first point is a problem that cannot be solved, since there is no hard truth about what people consider a shopping area or not. Most likely, user studies will have to be conducted in order to compare the areas found to the areas users think are part of a category. The second problem is a well known problem and relates to the data-source itself.

The question of how people communicate and discuss about space is an important aspect in spatial information theory (Weiser and Frank 2013). Especially when intending to compare settings in different cities and countries, cultural and language specific differences might pose challenges for processing the data. For example, in some countries people would relate the place description cafe to a small restaurant, whereas in other countries people could relate the term to coffee company brands. Comparing these different concepts is not directly possible. Therefore, there is a need for more research in the mapping of place affordances to semantics used in VGI.

An issue to be addressed in future work concerns the consideration of spatial relations among objects or categories of objects. While the presence of a certain type of object allows for affording a certain activity, the relative configuration of such objects in space also plays a role. Consider, for example, one is interested in identifying panoramic areas: the simple existence and the vicinity of a visually-appealing entity (e.g., a lake) and of a walkable and recreational area (e.g., a green spot with some benches) is not enough to categorize the area as panoramic spot. There might be a wall or building in between, hindering the line of sight going from the benches to the lake. Accordingly, for future work, we plan to integrate spatial configuration analysis (Fogliaroni 2013), so that finer differentiation between the settings is possible.

Concerning the method itself, in the future we will explore the possibility of extracting the characteristics of a defined setting, to create a description D that finds settings in other areas. For instance, in the presented use case (Sect. 4), Alice would be able mark an area on the map, from which the descriptions for the region growing algorithm is extracted and used to search for interesting areas in Vienna.

6 Summary and Conclusion

In this paper, we propose a novel approach to find and extract urban settings from *typed* point data. We were able to implement a framework that can be used for spatial search or analysis. We presented a formalization of our approach that is based on the idea of region growing—an Image Segmentation technique—described the implementation of it, and illustrated its feasibility by applying it to a use case scenario. In the case study, we show that our implementation enabled us to find well-known shopping areas in Vienna and London by using raw data of OpenStreetMap as a source. We analyzed the results by applying similarity-metrics that potentially enable a user to compare well-known shopping areas between cities. Built upon the preliminary findings, we identified various improvements and open questions, that once solved can lead to novel ways of searching, analyzing or comparing cities.

Acknowledgments We acknowledge the work of © OpenStreetMap contributors,[4] and Leaflet.[5] This research was partially funded by the Vienna University of Technology through the Doctoral College Environmental Informatics.

References

Abdalla, A., & Frank, A. U. (2012). Combining trip and task planning: How to get from a to passport. In *Geographic information science* (pp. 1–14). Berlin: Springer.

Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(6), 641–647.

Couclelis, H. (1992). Location, place, region, and space. *Geography's Inner Worlds, 2*, 15–233.

Couclelis, H., & Gale, N. (1986). Space and spaces. *Geografiska Annaler. Series B. Human Geography* (pp. 1–12).

D'Antonio, F., Fogliaroni, P., & Kauppinen, T. (2014). VGI edit history reveals data trustworthiness and user reputation. In *17th AGILE International Conference on Geographic Information Science (Short Paper)*.

Fan, J., Zeng, G., Body, M., & Hacid, M.-S. (2005). Seeded region growing: An extensive and comparative study. *Pattern Recognition Letters, 26*(8), 1139–1156.

Fogliaroni, P. (2013). *Qualitative spatial configuration queries: Towards next generation access methods for GIS*. AKA, Akad: Verlag-Ges.

Freundschuh, S. M., & Egenhofer, M. J. (1997). Human conceptions of spaces: Implications for GIS. *Transactions in GIS, 2*(4), 361–375.

Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal, 69*(4), 211–221.

Humayun, M. I., & Schwering, A. (2012). Representing vague places: Determining a suitable method. In *Proceedings of the International Workshop on Place-Related Knowledge Acquisition Research (P-KAR 2012), Monastery Seeon, Germany* (Vol. 881, pp. 19–25). Citeseer.

[4] http://www.openstreetmap.org/copyright.

[5] http://leafletjs.com.

Jones, C. B., Alani, H., & Tudhope, D. (2001). Geographical information retrieval with ontologies of place. In *Spatial information theory* (pp. 322–335). Berlin: Springer.

Jordan, T., Raubal, M., Gartrell, B., & Egenhofer, M. (1998). An affordance-based model of place in GIS. In *8th International Symposium on Spatial Data Handling, SDH* (Vol. 98, pp. 98–109).

Keßler, C., & de Groot, R. T. A. (2013). Trust as a proxy measure for the quality of volunteered geographic information in the case of OpenStreetMap. In *Geographic information science at the heart of Europe* (pp. 21–37). Berlin: Springer.

Lynch, K. (1960). *The image of the city* (Vol. 11). Cambridge: MIT Press.

Mark, D. M., & Frank, A. U. (1991). *Cognitive and linguistic aspects of geographic space* (Vol. 63). Berlin: Springer.

Meegan, R., & Mitchell, A. (2001). It's not community round here, it's neighbourhood: Neighbourhood change and cohesion in urban regeneration policies. *Urban Studies, 38*(12), 2167–2194.

Montello, D. R. (1993). Scale and multiple psychologies of space. In *Spatial information theory a theoretical basis for GIS* (pp. 312–321). Berlin: Springer.

Montello, D. R., Goodchild, M. F., Gottsegen, J., & Fohl, P. (2003). Where's downtown? Behavioral methods for determining referents of vague spatial queries. *Spatial Cognition and Computation, 3*(2–3), 185–204.

Mooney, P., & Corcoran, P. (2012). The annotation process in OpenStreetMap. *Transactions in GIS, 16*(4), 561–579.

Raubal, M. (2001). Human way finding in unfamiliar buildings: A simulation with a cognizing agent. *Cognitive Processing, 2*(3), 363–388.

Richter, D., Vasardani, M., Stirlng, L., Richter, K.-F., & Winter, S. (2013). Zooming in–zooming out hierarchies in place descriptions. In *Progress in location-based services* (pp. 339–355). Berlin: Springer.

Schatzki, T. R. (1991). Spatial ontology and explanation. *Annals of the Association of American Geographers, 81*(4), 650–670.

Scheider, S., & Janowicz, K. (2014). Place reference systems: A constructive activity model of reference to places. *Applied Ontology, 9*(2), 97–127.

Shimabukuro, Y. E., Batista, G. T., Mello, E. M. K., Moreira, J. C., & Duarte, V. (1998). Using shade fraction image segmentation to evaluate deforestation in landsat thematic mapper images of the amazon region. *International Journal of Remote Sensing, 19*(3), 535–541.

Smith, B. (1995). On drawing lines on a map. In: *Spatial information theory: A theoretical basis for GIS* (pp. 475–484).

Tuan, Y.-F. (1979). *Space and place: Humanistic perspective* (p. 1979). Berlin: Springer.

Vasardani, M., Winter, S., & Richter, K.-F. (2013). Locating place names from place descriptions. *International Journal of Geographical Information Science, 27*(12), 2509–2532.

Weiser, P., & Frank, A. U. (2013). Cognitive transactions: A communication model. In: T. Tenbrink, J. G. Stell, A. Galton, & Z. Wood (Eds.), *COSIT* (Vol. 8116, pp. 129–148). Lecture Notes in Computer Science. Berlin: Springer.

Winter, S., Kuhn, W., & Krüger, A. (2009). Guest editorial: Does place have a place in geographic information science? *Spatial Cognition & Computation: An Interdisciplinary Journal: Special Issue: Computational Models of Place*, *9*(3), 171−173.

Winter, S., & Truelove, M. (2013). Talking about place where it matters. In *Cognitive and linguistic aspects of geographic space* (pp. 121–139). Berlin: Springer.

Zielstra, D., & Zipf, A. (2010). A comparative study of proprietary geodata and volunteered geographic information for Germany. In *13th AGILE International Conference on Geographic Information Science*.

Central Places in Wikipedia

Carsten Keßler

Abstract Central Place Theory explains the number and locations of cities, towns, and villages based on principles of market areas, transportation, and socio-political interactions between settlements. It assumes a hexagonal segmentation of space, where every central place is surrounded by six lower-order settlements in its range, to which it caters its goods and services. In reality, this ideal hexagonal model is often skewed based on varying population densities, locations of natural features and resources, and other factors. In this paper, we propose an approach that extracts the structure around a central place and its range from the link structure on the Web. Using a corpus of georeferenced documents from the English language edition of Wikipedia, we combine weighted links between places and semantic annotations to compute the convex hull of a central place, marking its range. We compare the results obtained to the structures predicted by Central Place Theory, demonstrating that the Web and its hyperlink structure can indeed be used to infer spatial structures in the real world. We demonstrate our approach for the four largest metropolitan areas in the United States, namely New York City, Los Angeles, Chicago, and Houston.

1 Introduction

Central Place Theory was developed in the 1930s, following the observation of recurring patterns in the arrangement of settlements of different sizes (Christaller 1933; Baskin 1966).[1] It explains the number and locations of cities, towns, and

[1]The title of the current paper alludes to the title of the original publication introducing Central Place Theory by Christaller (1933), translated by Baskin (1966): *Central Places in Southern Germany*.

C. Keßler (✉)
Center for Advanced Research of Spatial Information and Department of Geography, Hunter College, City University of New York, New York, USA
e-mail: carsten.kessler@hunter.cuny.edu

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_3

villages based on principles of market areas, transportation, and socio-political interactions between settlements. Under perfect—and somewhat unrealistic—conditions, Central Place Theory predicts a hexagonal segmentation of space, such that six lower-order settlements (e.g., towns) arrange around one higher-order settlement (e.g., a city). These purely spatial explanations of Central Place Theory were later extended to take economic considerations, such as competition, into account (Lösch 1954). Sizes and market areas of the respective settlements are not fixed, but rather depend on population density, locations of natural features and resources, and other factors.

Central Place Theory has been shown to apply in a number of places, especially when the local situation is close to the underlying assumptions of the theory (Berry and Garrison 1958b; Brush 1953, for example). Even in cases where the spatial arrangement of the settlements cannot be easily explained by Central Place Theory, the formation of lower-order settlements around central places that provide certain goods or services is still evident and can be observed everywhere in the developed world. As such, Central Place Theory explains networks of dependencies, where smaller settlements depend on goods, services, and the job markets of a larger settlement in their vicinity.

The premise of the research presented here is that those central places and the spatial configuration of settlements in their range can be inferred from the link structure on the Web. Using a corpus of georeferenced documents from the English language edition of Wikipedia, our results indicate that these dependencies between smaller and larger settlements are reflected in the number of references between their corresponding Wikipedia pages. The underlying assumption is that a central place will be referred to more often, specifically from places in its range that are functionally dependent.

Following this approach, we assess the range of a central place based on the frequency distribution of the distances to *referring* places, i.e., places whose corresponding Wikipedia pages link to this place, or mention it in the text. We assign weights to the incoming links from other places based on the count of references on their pages; for example, the Wikipedia page for Jersey City, New Jersey, contains one hyperlink to the page for New York City, and 9 mentions of the term "New York City" in its text, resulting in a total of 10 references from Jersey City to New York City. In comparison, the page for Hoboken, New Jersey, contains 4 hyperlinks and 49 mentions, resulting in a total of 53 references. We show how these reference counts can be employed as weights in our model to account for the relative importance (or *centrality*) of New York City for Jersey City and Hoboken, respectively.

Like most other studies that employ the Web as a source for geographic information, we face the Geoweb Scale Problem (Hecht and Moxley 2009): The geometry of each place is only available as a point coordinate in Wikipedia, independent of whether the represented feature is as small as a statue in Central Park, or as large as New York State. Therefore, the semantics of the relationships between two places plays an important role, in addition to using the number of references for the weighting. We take into account the administrative hierarchy

between places obtained from the GeoNames gazetteer[2] and the DBpedia ontology (Lehmann et al. 2012) as a filter to tackle this problem. We demonstrate the general feasibility of this combined approach by analyzing the link structure of the four largest regional capitals in the United States, namely New York City, Los Angeles, Chicago, and Houston.

The remainder of this paper is organized as follows: In the next section, we discuss relevant related work on Central Place Theory and on the analysis of Wikipedia contents. Section 3 describes the process of obtaining and preparing the dataset used in this study, followed by a specification of the main characteristics of the dataset in Sect. 4. The process of identifying central places in our dataset is introduced in Sect. 5, including a detailed discussion of the different choices made in the process. Section 6 discusses the obtained results, followed by concluding remarks in Sect. 7.

2 Related Work

This section reviews the core ideas of Central Place Theory and gives an overview of relevant related work on the analysis of geospatial content on the Web, with a focus on Wikipedia.

2.1 Central Place Theory

Central Place Theory (Christaller 1933; Baskin 1966) explains the spatial configurations of central places from a purely economic perspective, viewing them mainly as locations where people come together to trade goods and services. The theory is based on a number of assumptions, such as an isotropic plane, evenly distributed population and resources, profit-oriented sellers, and economic customers who aim to minimize their travel to obtain goods. Lösch (1954) later relaxed this rigid economic perspective, modifying the theory to optimize for consumer welfare. The rank order of central places distinguishes hamlets (first-order centers), villages (second-order centers), towns (third-order centers), cities (fourth-order centers), and regional capitals (fifth-order centers). Figure 1 shows the hexagonal spatial configuration arising from these five orders of centers. According to this categorization, all four places that we investigate here are regional capitals, and we try to identify the depending cities and towns around them.

Central Place Theory has been studied and evaluated from a number of perspectives since the 1930s. Berry and Garrison (1958b) performed a detailed analysis of central places in Snohomish County, Washington. They found the predictions of

[2] http://www.geonames.org.

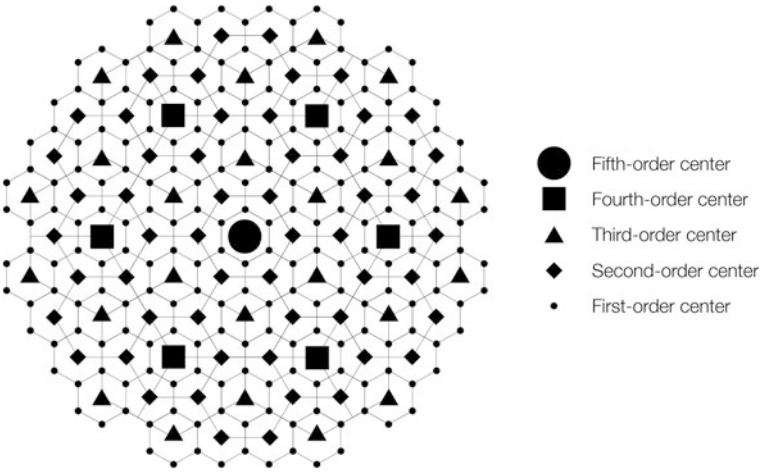


Fig. 1 Hexagonal spatial configuration of the five order central place system. Adapted from Openshaw and Veneris (2003)

Central Place Theory largely confirmed by the data collected, with the exception of a small number of places that had seen a recent increase in population. They also used the data collected during this study to show that Lösch's idea of an economic equilibrium between places does not hold below certain population densities or levels of urbanization (Berry and Garrison 1958a). Hsu (2012) later showed that the city sizes in Central Place Theory can be formalized using a power law model. Openshaw and Veneris (2003) evaluated the expected trip distributions in a central place model against spatial interaction models, finding that most spatial interaction models were unable to produce the trip distributions predicted by Central Place Theory.

2.2 Geographic Analyses of the Web and Wikipedia

Both Wikipedia and the Web as a whole have been employed in a number of ways to answer different kinds of geographic research questions. Frankenplace (Adams and McKenzie 2012) is an online application that extracts and analyzes qualitative geographic information from travel blogs, turning it into a data source for similarity-based place search. In previous work, we have demonstrated that the shapes of real-world features can be approximated based on geotagged photos on the Web (Keßler et al. 2009). Gao et al. (in press) show how Volunteered Geographic Information can even be used for the construction of gazetteers. Salvini (2012) applies spatialization techniques to the English language edition of Wikipedia in order to analyze the functional structure of the global network of cities.

Likewise, the spatial aspects of Wikipedia have already been the focus of different studies. Takahashi et al. (2011) present an approach to extract the significance of spatio-temporal events from Wikipedia. Their idea of links as impact propagation is similar to our approach that uses the number of references as an indicator of centrality of a place. Concerning the locations of contributors to Wikipedia, Lieberman and Lin (2009) show that the geographic coordinates of pages edited by a user often cluster tightly. Hecht and Gergle (2010) show, however, that this *localness* does not generally apply to any kind of Volunteered Geographic Information. Hecht and Moxley (2009) have conducted an extensive experiment to demonstrate the validity of Tobler's First Law of Geography (Tobler 1970) in Wikipedia across different language editions of the online encyclopedia. They have shown empirically that places that are closer to each other in geographic space are also more likely to be related—i.e., interlinked—on Wikipedia.

In this paper, we take the concept of relatedness one step further by investigating how strong a place relates to another one, using the number of links and mentions as indicators. We show that this degree of relatedness reflects the functional dependencies between places as explained by Central Place Theory.

3 Data Access and Preprocessing

The dataset used in this study has been limited to a bounding box spanning the area between 50°N, −128°W (west of Vancouver Island, British Columbia, Canada) and 25°N, −64°W (north of Puerto Rico). It thus contains all of the contiguous United States as the focus area of our study. We limited the dataset under consideration to this area to have a consistent dataset in terms of language, taking into account only articles from the English language version of Wikipedia.[3] Previous research has shown that user generated content is not as local as the premise of Volunteered Geographic Information (Goodchild 2007) suggests, especially in the case of Wikipedia (Hecht and Gergle 2010). Nonetheless, using this combination of geographic area and Wikipedia articles in the main language spoken in this area should avoid the introduction of inconsistencies that arise from crossing language barriers. We hence defer the investigation of potential differences across geographic regions and language barriers to future research. Finally, this dataset is still tractable enough in terms of overall size in order to explore the feasibility of the general idea of this paper.

Since Wikipedia itself does not support queries by location through its API,[4] we used DBpedia instead. DBpedia (Lehmann et al. 2012) provides facts extracted from Wikipedia as structured Linked Open Data (Berners-Lee 2009). Among the facts extracted from Wikipedia are the geocordinates that are provided at the top

[3] http://en.wikipedia.org.

[4] https://www.mediawiki.org/wiki/API.

right of a page for many subjects that have a geographic location. The coordinates are represented using the W3C Basic Geo Vocabulary (W3C Semantic Web Interest Group 2004). This allowed us to retrieve all English language Wikipedia pages and their geographic coordinates within our bounding box, using DBpedia's SPARQL (Harris and Seaborne 2013) endpoint.[5] The result was fed into a *places* collection stored in a local MongoDB[6] instance, consisting of entries of the following form:

```
{ "_id"  : ObjectId("5466a15e080cb903020330fe"),
  "loc"  : { "type" : "Point",
             "coordinates" : [ -73.99028015136719,
                               40.62472152709961 ] },
  "page" : "http://en.wikipedia.org/wiki/Brooklyn" }
```

This collection of all georeferenced Wikipedia pages was used in the next step to download the actual contents of each page, using the XML export function of the Wikipedia API. The XML export is more straight-forward to parse than the actual HTML pages, while providing the same information. Each XML document was parsed for links to other georeferenced Wikipedia pages listed in our places collection, as well as for further *mentions* of such linked pages. As an example, when parsing the contents of the page for http://en.wikipedia.org/wiki/Hoboken,_New_Jersey, we will find links to the page for New York City. In Wiki syntax, this is represented as `[[New York City]]`, and rendered as

```
<a href="http://en.wikipedia.org/wiki/New_York_City">New York City</a>
```

by the MediaWiki engine driving Wikipedia. If we find such a link, we also scan the whole page for any other occurrences of the words *New York City* (without a hyperlink), since it is common practice to only link the first occurrence of a subject to its Wikipedia page, and not every single one. Taking these mentions into account gives us a more detailed impression of how often a georeferenced page is being referred to from other pages. Links of the form `[[Washington, D.C.|Washington]]`, that provide a different text to be shown (*Washington* in this example), are taken into consideration the same way.

By parsing all pages in this fashion, we built a collection of *links*, covering all pairs of pages that link to each other. One link consists of the linking page (*from*), the linked page (*to*), the number of actual *links*, the number of *mentions*, and the geographic *distance* (in meters) between the two georeferenced pages,[7] calculated from their respective coordinates in the *pages* collection:

[5] http://dbpedia.org/sparql.

[6] http://mongodb.org.

[7] The distance has been calculated as great circle distance assuming a spherical Earth. The errors introduced by this simplification should be negligible in the context of this study.

```
{ "_id"      : ObjectId("54831bd6080cb9001a09ebb3"),
  "from"     : "http://en.wikipedia.org/wiki/Hoboken,_New_Jersey",
  "to"       : "http://en.wikipedia.org/wiki/New_York_City",
  "links"    : NumberLong(4),
  "mentions" : NumberLong(49),
  "distance" : 4618.070713194219 }
```

4 Dataset Characteristics

As of November 14, 2014, the dataset we retrieved in the way described in the previous section consists of 242,896 georeferenced subjects in the English language edition of Wikipedia. Parsing the pages' contents extracted 1,517,772 unique combinations of referring (*from*) and referred places (*to*), each with information on counts of links and mentions, and the distance between the respective pages as outlined in the previous section. The majority of those linking pages only contain a single link to the referenced page, as shown in Table 1. On average, a linking page contains ~ 1 link and ~ 0.86 textual mentions of the referred page. The maximum values observed are 28 links and 100 mentions, respectively, both of which are reached by multiple links. The average link is between places that are ~ 264 km apart, whereas half of all links in our collection are between places that are ~ 21 km or less apart. The biggest distance crossed by any of the links in our collection is ~ 5496 km, which is about 600 km short of the diameter of our bounding box.

Figure 2 shows the frequency distribution for the distances of each link. For this bar plot, the collection of all links has been divided into 100,000 subsets based on distance bins, each 54 m wide.[8] Each bar in the chart shows the number of links in this bin, starting from 0 to 54 m bin, at the very left, followed by $55 - 109$ m bin, etc. The orange bars show the number of pages linking to each other within the bin, while the gray bars are weighted by the number of references: For example, let place A and B be 300 m apart, and the page for place A links to the page for place B 3 times, and contains 2 more mentions. This would result in a single count in $270 - 324$ m bin for the unweighted bars (orange), and in 5 counts for the weighted bars (gray).

The bar plot clearly shows the expected power-law distribution, i.e., places closer to each other link to each other more often, and they also *mention* each other more often in the text. Note that Fig. 2 only shows the top 1 % of the whole distribution, i.e., the number of links converges very quickly towards 1, and the number of mentions towards 0. We will make use of this fact in our detection of central places in Sect. 5. Figure 3 confirms this at the individual level, showing the

[8] While the choice of the number of bins is arbitrary, comparable results have been obtained with different bin sizes.

Table 1 Overview of the link collection

	Links	Mentions	Distance (m)
Min	1	0	0
Median	1	1	21,254
Mean	1.035	0.8554	264,972
Max	28	100	5,496,277

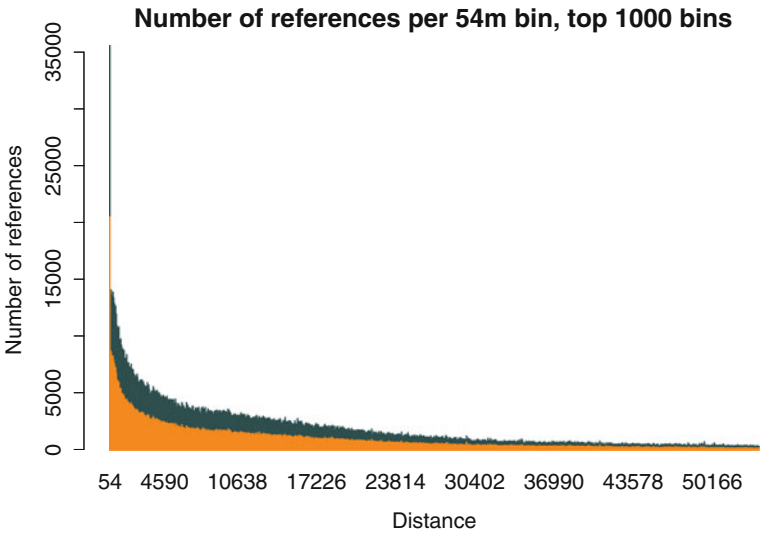


Fig. 2 Number of references in each 54 m bin, weighted (*gray*) versus unweighted (*orange*). Note that these are *not* stacked to reflect that the unweighted links are included in the weighted references. The chart is limited to the top 1 % of all bins, i.e., the rightmost bin ends at 54 km

total number of references (links and mentions) against the distance for all ~ 1.5 million links: Very high numbers of references can only be observed between places that are spatially close to each other, while the vast majority of links only contains a small number of references.

Table 2 shows the ten places with the highest number of incoming references, i.e., these are the most linked-to and mentioned places in our collection. Unsurprisingly, it consist of large-scale administrative units, led by the United States with close to to 200,000 incoming references. Several US states lag behind at about 30,000 references. The first cities in this ranking are New York City at rank 17 (8647 references), Chicago at rank 19 (6892 references), and Washington, D.C. at rank 24 (4718 references). While the order of this ranking is hardly surprising, it shows that these large-scale administrative units need special handling during our identification of central places.

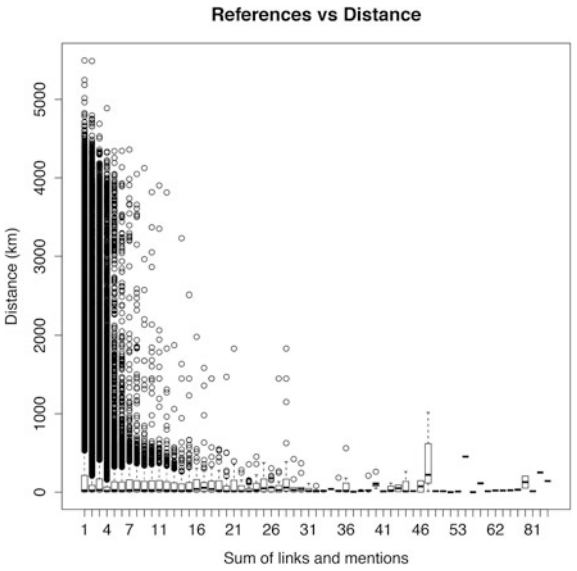


Fig. 3 *Boxplot* showing the variance in distances between places, grouped by total number of references (links and mentions) for all 1.5 million links

Table 2 Overview of the top referenced places (links plus mentions)

	Place	References
1	http://en.wikipedia.org/wiki/United_States	199,605
2	http://en.wikipedia.org/wiki/California	35,990
3	http://en.wikipedia.org/wiki/Ohio	29,598
4	http://en.wikipedia.org/wiki/New_York	28,928
5	http://en.wikipedia.org/wiki/Illinois	27,600
6	http://en.wikipedia.org/wiki/Wisconsin	27,125
7	http://en.wikipedia.org/wiki/Indiana	26,893
8	http://en.wikipedia.org/wiki/Texas	25,132
9	http://en.wikipedia.org/wiki/Florida	20,981
10	http://en.wikipedia.org/wiki/Kentucky	17,776

5 Analyzing Central Place Structures

This section introduces an approach to analyze central place structures in Wikipedia based on the dataset discussed in Sect. 4. Using the four largest cities in the United States as case studies, we discuss the influence of weights and semantic aspects of the approach.

5.1 General Approach

The premise of this research is that link structures in Wikipedia reflect real-world dependencies of smaller settlements (e.g., towns) on a central place (e.g., a city). We hence interpret every link as a pointer to a central place, where (a) the number of references in that specific link reflects the degree of dependency between the respective places, and (b) the total number of incoming references reflects the centrality—or relative importance—of a place. Following these assumptions, we can reveal the structure around a central place P as follows:

- Retrieve all links pointing to P.
- Remove all links that are beyond a weighted distance D, since many places have incoming links from places for away, where no spatial interaction in the sense of Central Place Theory is given (e.g., references between the pages of partner cities). This step is crucial and will be discussed in more detail in Sect. 5.2.
- Generate the convex hull of all remaining links to represent the range of P.
- For every remaining link, inspect the linking place and calculate its own relative importance based on its number of incoming links.
- Remove all links from this list where the linking place either has an administrative relationship to P, is not a settlement (see Sect. 5.3), or where the place has already been added to the structure of P in a previous iteration.
- From the remaining candidates, keep the top 6 closest places I according to their weighted distance. The number 6 follows from the hexagonal segmentation of space underlying Central Place Theory.
- Iteratively repeat the process for each place I to reveal the structure at the next lower order. Iteratively repeat the process for each place I to reveal the structure at the next lower order.

This approach will yield the 6 most relevant settlements at the next lower order; e.g., if P is a regional capital (fifth-order center), the first iteration will yield 6 cities (fourth-order center) in the range of P. The second iteration will yield the 6 most relevant towns (third-order centers) in range of each of those 6 cities, yielding a total maximum of 36 towns. Some of these 36 places will most likely appear twice at the same order, i.e., a third-order place may be in the range of two second-order places. Under perfect conditions, this would yield 24 unique third-order places, as every second-order place shares 4 third-order places in its range with another second order place (see Fig. 1).

5.2 Distance and Weighting Considerations

Since many places in our collection receive incoming references from places all over the US, a mechanism is required to reliably generate a realistic convex hull representing the place's range. It needs to weigh the distance crossed by a link from

a lower-order to a higher-order place against the degree of dependence. As discussed in Sect. 1, we use the number of references as an indicator of dependence, i.e., the higher the number of references, the more dependent a place is. We use a naïve approach here, where the weighted distance d_w for a link is defined as the geographic distance d divided by the number of references r:

$$d_w = \frac{d}{r}$$

This approach causes lower-order places with a high number of references to be drawn towards the higher-order place, figuratively speaking. If a place A is twice as far away as a place B from a higher-order place P, they would be assigned the same weighted distance if A has twice as many references to P as B.

Using this weighting approach, we generate the input for the actual selection of the referring places we want to accept as being in the range of a place P. Figure 4 gives an overview of the 10 %–100 % quantiles for the case of New York City, with 10 % increments (the smallest area in the right part of Fig. 4 is the convex hull for the 10 % quantile, the next larger one for the 20 % quantile, etc.). We have experimented with different quantiles and found that taking into account the 75 % quantile of all weighted links referring to a place yields the most realistic results in the case of the four metropolitan areas under consideration (see Sect. 6 for a more detailed discussion).

5.3 Semantic Aspects

Large administrative areas tend to receive a high number of references, as shown in Table 2. Only relying on the 75 % quantile of weighted distances would hence weave these administrative units into our central place structure. In most cases, these references would be meaningless for our purpose, though; the state of Illinois as a geographic entity does not contribute anything to the centrality of Chicago. It is rather the settlements *within* Illinois that interact with Chicago, and bear its importance in the central place structure. The same goes for counties and the federal state. Likewise, very small "places"—in the sense of some real-world entity that has a georeferenced Wikipedia page—are not meaningful in our structure. Parks, buildings, or companies should not be reflected in our structure, even if they are within a place's range, and their number of incoming references indicates centrality.

We make use of (a) the administrative place hierarchy and (b) information about the types of things we are looking at to decide whether to include a candidate in the central place structure or not. Places are only included in the structure if:

- They are at the same or a lower level in the administrative hierarchy as the place P under consideration, according to GeoNames.

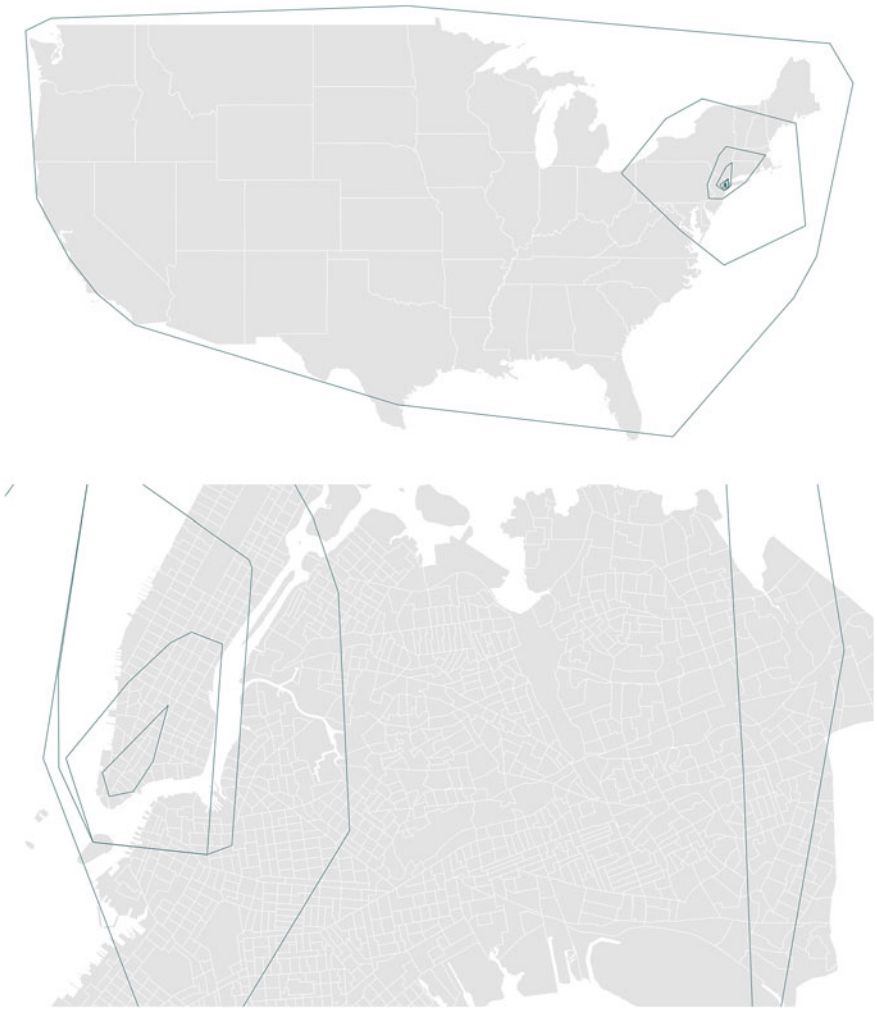


Fig. 4 Different quantiles of weighted differences. Every convex hull represents 10 % quantiles of the incoming weighted links for New York City. When all links are taken into account, the convex hull contains the entire lower 48 states (*top*), whereas the 10 % quantile contains only a few blocks in Lower Manhattan (*bottom*)

- They are not a child of P in the administrative hierarchy, according to GeoNames (e.g., Brooklyn would be excluded if we are looking at New York City, as it is one of the city's five boroughs and hence a child in its hierarchy).
- They are of type settlement[9] (including any subtypes), as defined in the DBpedia ontology (Lehmann et al. 2012).

[9]See http://dbpedia.org/ontology/Settlement.

The following section evaluates the results obtained using our methodology for New York, Los Angeles, Chicago, and Houston.

6 Evaluation

This section evaluates the approach, looking at the results obtained for the four largest metropolitan areas in the US. For all figures in this section, black is used for fifth-order centers (i.e., New York City, Los Angeles, etc.), red for fourth-order centers, and orange for third-order centers. Meaningful results for the second- and first-order centers could not be obtained, due to the very low number of incoming links to the pages for the identified third-order centers.

Figure 5 shows the four central place structures evaluated, using the 75 % quantile of the weighted links between places. The difference in scale between the four areas is evident. While we have not conducted a systematic evaluation against population density, it seems like structures yielded in higher population density areas are more compact. This finding needs to be confirmed by taking into account more examples in the future, and by evaluating the results against population density data.

Beverly, Massachusetts is incorrectly assigned as a third-order center to Beverly Hills, California, due to the high number of references to Beverly Hills on its page. This is an outlier that is not handled well by our approach yet. Beverly Hills,

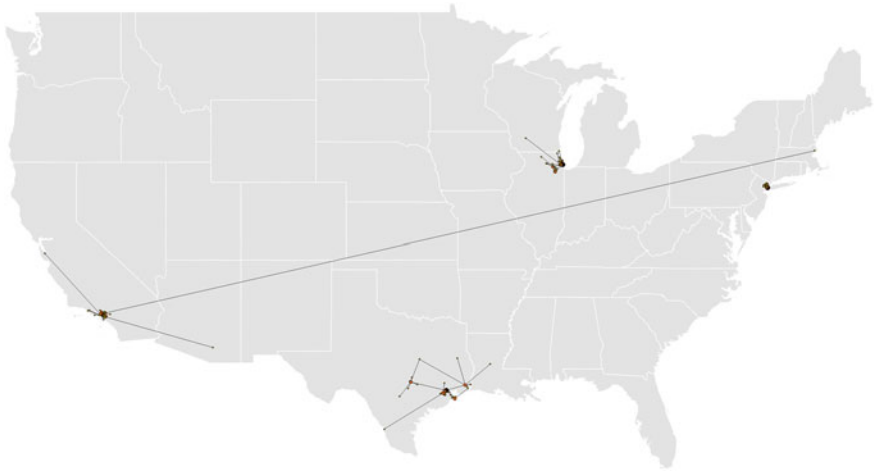


Fig. 5 Overview of the four central place structures evaluated, using the 75 % quantile of the weighted links between places

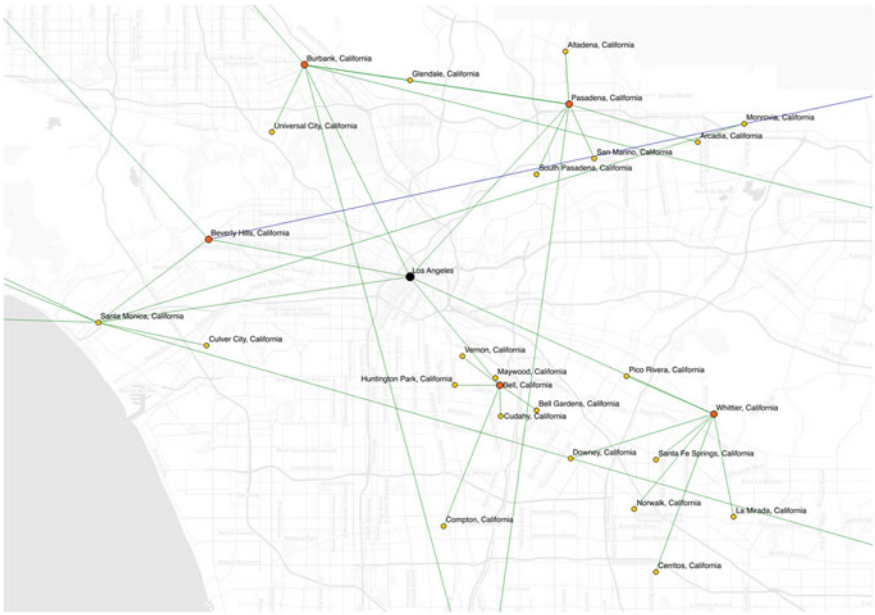


Fig. 6 Central place structure for Los Angeles

however, does show some other uncommon properties, namely that we could only identify 3 linking places outside of its own administrative hierarchy, one of them being Beverly on the East Coast. This low number of incoming links from the vicinity may point to a very separate, almost *gated* community. Independent of the underlying reasons, such outliers could be handled by a "hard" geographic cutoff distance, beyond which places are not taken into consideration any more. This would also speed up the computation of the central place structures when automating this approach further and expanding it to the whole world.

All four structures face away from the water, which intuitively makes sense, but prevents a meaningful comparison with the structures predicted by Central Place Theory. Some of the fourth-level centers, however, approach a hexagonal configuration of space, such as Bell and Pasadena in Fig. 6, or Montclair in Fig. 7. Using the underlying map as an indicator for population density again, the lower-order centers generally seem to lean towards areas with higher population areas. This is also to be expected, but needs further investigation. The administrative hierarchy of New York City also heavily influences the results shown. Since Queens, Brooklyn, and the Bronx were among the most central places linking to New York City, but excluded because of their being part of New York City, the whole central place structure faces towards New Jersey.

Fig. 7 Central place structure for New York City

Both the structures for Houston and Chicago (Fig. 8) show a small number of links that span long distances, which is somewhat unexpected given the high population densities in these areas. The number of references retrieved from Wikipedia indicate that many places in Texas are not as well documented as the New York City or Los Angeles areas, for example. This may explain these unexpected results, as our method strongly relies on a reasonably detailed input. As an example, when processing the structure around Bellaire, Texas, we only identified one third order place linking to it within the 75 % quantile.

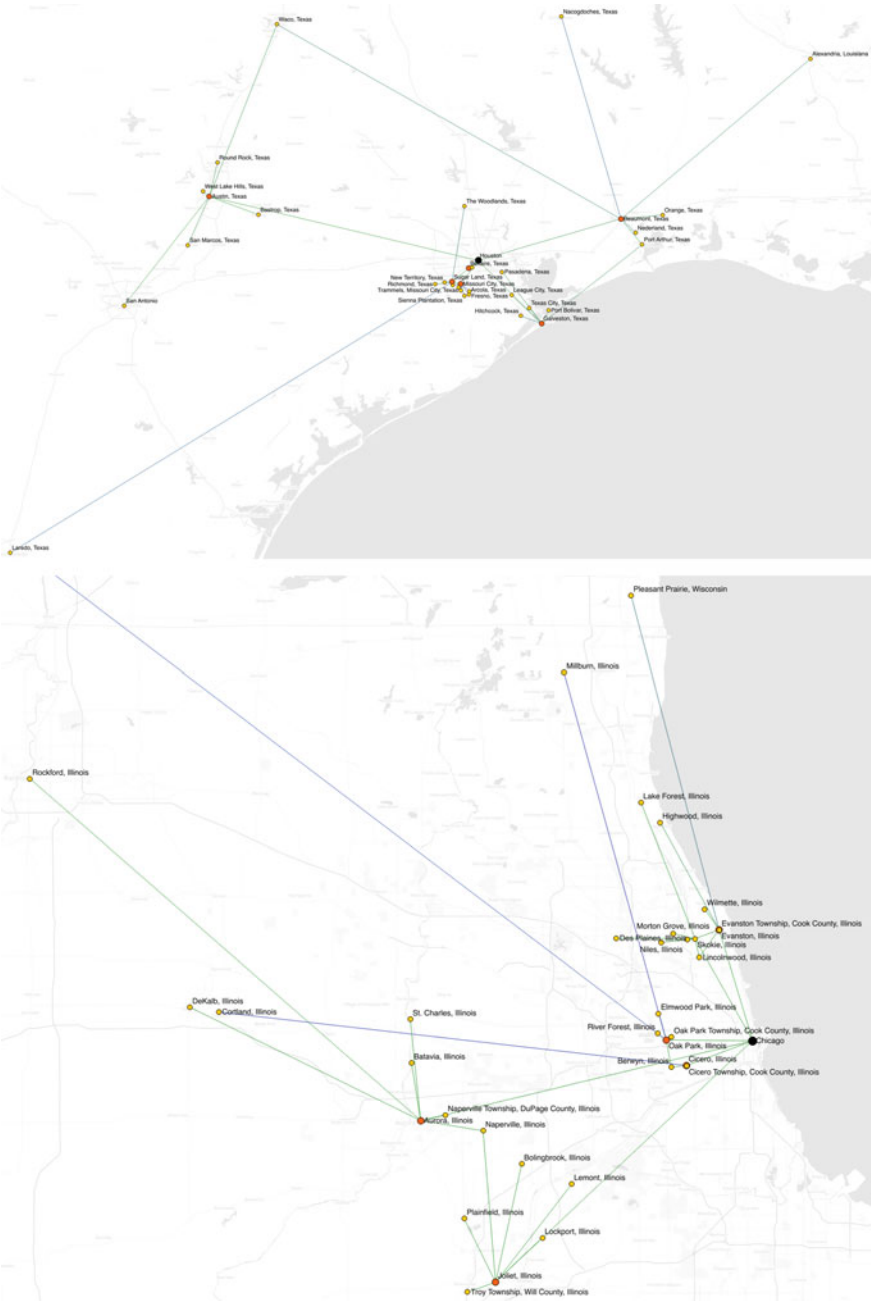


Fig. 8 Central place structure for Houston (*top*) and Chicago (*bottom*)

7 Conclusions

We have introduced an approach to extract the structure around a central place and its range from the link structure in the English language edition of Wikipedia. Using weighted distances and semantic annotations, we have demonstrated that the Web and its hyperlink structure can indeed be used to infer spatial structures in the real world. While the results vary significantly depending on population density and natural features—all places considered in the paper are near the sea or a large lake—parts of the identified structures match the predictions of Central Place Theory well. The presented results are only a first indication that the Web does not only exhibit patterns of spatial autocorrelation (Hecht and Moxley 2009) and the shapes of real-world features (Keßler et al. 2009), but it also reflects interactions between places. The study indicates that the link structure on the Web mirrors which places functionally depend on each other, and to what degree. Our results also point to the fact that the link structure in Wikipedia is only useful down to the third-order centers, as these are usually already small towns whose Wikipedia pages do not have any significant numbers of incoming links.

While the structures around the regional capitals investigated in this paper seem intuitive, the results clearly need a more thorough, quantitative analysis, also in order to fine-tune the process of identifying the structures. With the software tools built for this process, the next step will be to fully automate the generation of central place structures from Wikipedia. This will allow us to experiment with different variations of the approach, and run it on a larger input body.

Acknowledgments The base maps used in Figs. 6, 7, and 8 have been provided by Stamen Design[10] under a Creative Commons License.[11] The maps are based on OpenStreetMap data, provided under the Open Database License.[12]

References

Adams, B., & McKenzie, G. (2012). Frankenplace: An application for similarity-based place search. In: *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*.

Baskin, C. W., (1966). *Central places in Southern Germany*. Englewood Cliffs, NJ: Prentice Hall.

Berners-Lee, T., (2009). *Linked data—design issues*. Online: http://www.w3.org/DesignIssues/LinkedData.html.

Berry, B. J. L., & Garrison, W. L. (1958a). A note on central place theory and the range of a good. *Economic Geography, 34*(4), 304–311.

Berry, B. J. L., & Garrison, W. L. (1958b). The functional bases of the central place hierarchy. *Economic Geography, 34*(2), 145–154.

[10] http://stamen.com.

[11] http://creativecommons.org/licenses/by/3.0.

[12] http://www.openstreetmap.org/copyright.

Brush, J. E. (1953). The hierarchy of central places in Southwestern Wisconsin. *Geographical Review, 43*(3), 380–402.

Christaller, W. (1933). *Die zentralen Orte in Süddeutschland*. Jena: Gustav Fischer.

Gao, S., Li, L., Li, W., Janowicz, K., & Zhang, Y. (2014). Constructing gazetteers from volunteered big geo-data based on Hadoop. *Computers, Environment and Urban Systems*. doi:10.1016/j.compenvurbsys.2014.02.004.

Goodchild, M. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal, 69* (4), 211–221.

Harris, S., Seaborne, A. (2013). SPARQL 1.1 query language. W3C *Recommendation.* Available from http://www.w3.org/TR/sparql11-query/.

Hecht, B., & Moxley, E. (2009). Terabytes of tobler: Evaluating the first law in a massive, domain-neutral representation of world knowledge. In: K. Hornsby, C. Claramunt, M. Denis, & G. Ligozat (Eds.), *Spatial information theory . Lecture Notes in Computer Science* (Vol. 5756, pp. 88–105). Heidelberg: Springer. doi:10.1007/978-3-642-03832-7_6.

Hecht, B. J., & Gergle, D. (2010). On the localness of user-generated content. In *Proceedings of The 2010 ACM Conference on Computer Supported Cooperative Work* (pp. 229–232).

Hsu, W. T. (2012). Central place theory and city size distribution. *The Economic Journal, 122* (563), 903–932.

Keßler, C., Maué, P., Heuer, J. T., & Bartoschek, T. (2009). Bottom-Up gazetteers: Learning from the implicit semantics of geotags. In: K. Janowicz , M. Raubal, S. Levashkin (Eds.), *Third International Conference on Geospatial Semantics* (GeoS 2009), Springer Lecture Notes in Computer Science 5892, pp. 83–102.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., et al. (2012). DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal, 1*, 1–5.

Lieberman, M. D., & Lin, J. (2009). You are where you edit: Locating Wikipedia contributors through edit histories. In *Proceedings of the Third International Conference on Weblogs and Social Media*, (ICWSM 2009), San Jose, California, USA, 17–20 May 2009.

Lösch, A. (1954). *The Economics of Location*. New Haven, CT: Yale University Press.

Openshaw, S., & Veneris, Y. (2003). Numerical experiments with central place theory and spatial interaction modelling. *Environment and Planning A, 35*(8), 1389–1404.

Salvini, M. M. (2012). Spatialization von nutzergenerierten Inhalten für die explorative Analyse des globalen Städtenetzes. PhD thesis, University of Zurich.

Takahashi, Y., Ohshima, H., Yamamoto, M., Iwasaki. H., Oyama, S., & Tanaka, K. (2011). Evaluating significance of historical entities based on tempo-spatial impacts analysis using wikipedia link structure. In *Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia*, (pp 83–92).

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography, 46*, 234–240.

W3C Semantic Web Interest Group. (2004). Basic geo (WGS84 lat/long) vocabulary. Online: http://www.w3.org/2003/01/geo/.

Applications of Volunteered Geographic Information in Surveying Engineering: A First Approach

Ioannis Sofos, Vassilios Vescoukis and Maria Tsakiri

Abstract Volunteered Geographic Information (VGI) has been used in various scientific domains and applications. Surveying Engineering is a field that has not yet exploited concepts like data and service sharing and reuse. This paper aims to suggest a framework that will support data sharing in Surveying Engineering by creating an online spatio-temporal information repository for land surveying projects. A data model to meet the needs for Surveying Engineering applications and accuracy requirements is introduced to facilitate the sharing of VGI information among Surveying Engineers. A fully functional prototype system has been developed and used to apply the proposed methodology in a large scale study undertaken by the Greek Ministry of Culture which involves the mapping of the historic center of Athens as part of the Archaeological Cadastre project. Results coming from data analysis indicate a substantial ($\sim$60 %) error reduction and also significant productivity raise ($\sim$25 %), while at the same time, the collected data are structured and saved in an online database, accessible by community users—professional Surveying Engineers who can in turn contribute to further improve the available data and services according to the principles of VGI applications.

Keywords Volunteered geographic information · Engineering applications · Spatial data and service sharing

I. Sofos (✉) · M. Tsakiri
National Technical University of Athens (NTUA), Athens, Greece
e-mail: gs.sofos@gmail.com

M. Tsakiri
e-mail: mtsakiri@central.ntua.gr

V. Vescoukis
Swiss Federal Institute of Technology Zurich, Zurich, Switzerland
e-mail: vvescoukis@ethz.ch

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_4

1 Introduction

Information sharing and reuse that has been made possible on the Internet has largely revolutionized many activities ranging from research to daily life activities over the past few years (Agrawal 2003; Ozyer et al. 2012). Global geo-spatial applications motivate the development of communities that share all kinds of geographic information, organized in local or even global data collections (Pinto and Jeffrey 2000). In 2007, the term "Volunteered Geographic Information (VGI)" was introduced and since then it has been used in a number of applications, including navigation and mapping (Dobson 2013). VGI has been characterized as "a special case of the more general Web phenomenon of user-generated content" (Goodchild 2011), and has been widely used in various fields. Two of the most successful and popular projects that rely on VGI are OpenStreetMap (OSM) and WikiMapia. The idea that has been successfully implemented in these projects is that mass geographic data coming from various sources, collected and assessed heterogeneously and with no central control, are aggregated in collections that anybody can access and process in order to deliver new geo-spatial products or services.

Surveying Engineers traditionally work on the field, collecting data in order to determine the shape and the positions of points on the earth's surface along with features of interest (Moffitt 1998). The surveying measurements are obtained with high accuracy instruments (Total Stations) that measure angles and distances. Other methods like Global Navigation Satellite Systems (GNSS) as well as techniques based on image processing (photogrammetry, remote sensing etc.) can also be applied on some cases. However, data of high accuracy produced in these processes usually remain private and isolated within the context they were produced in. The surveying engineering community seems to ignore the enormous potential of sharing such data, which can significantly improve the productivity and reduce the cost of the field measurement process.

This paper discusses the concept of VGI in surveying engineering applications. In particular, focus is placed on land surveying applications using Total Stations (TS) for data collection, which is the most common approach (Ghilani and Wolf 2008). Each point is spatially determined using measurements of distance and angle relative to known CPs that define a reference network. A typical workflow includes the establishment of the reference network, the acquisition of raw measurements in the field, as well as and the post-processing of measurement data, either by software embedded in the instruments or by desktop software. Whatever the case, this process is based on the following concepts: (a) every measurement station is an autonomous working node and (b) the measurement workflow is comprised of two discrete, sequentially executed steps, namely measurement and processing. Regardless of whether the processing is done by embedded software on-the-field or later, there is no dynamic interaction between the acquisition and the processing of measurements. As "dynamic interaction", is considered any decision-making activity that can impact either the measuring workflow itself (such as decisions on

what to measure next), or the validation of the accuracy of measurements. As a result, errors that might come up during post-processing can only be resolved by either backtracking or even re-visiting the field for new measurements.

Furthermore, as mentioned earlier, after processing is done, raw data are put on the side and practically rendered useless: it remains in digital files, in most cases without any standards-based structure or meta-data that could make them re-usable by other users, except possibly by those who originally acquired the measurements. The same stands for other related data types that support mapping processes, such as digital photos or related public records. The above remark is of special interest considering that surveying of a particular area can be performed several times by different engineering teams and/or on behalf of different employers, etc. However, re-usability of collected data is of major importance, as it would lead to multiple benefits, such as:

- Productivity boost and cost reduction due to re-using existing and validated available information.
- Accuracy improvement due to additional data that will be made available for processing.
- Filtering of erroneous observations or measurements.
- Possibility for temporal analysis of data for the same geographical area.
- Enabling of new applications that can make use of the data, possibly combined with other VGI openly available for the same geographical area.

The huge leaps of digital data processing and communications technologies during the past few years, combined with the globally available wireless communication networks, enable the development of any kind of application that shares common data and services, forming a cloud computing structure (Rimal and Lumb 2009).

These developments today enable the creation of a network for sharing surveying measurement data for engineering applications. Portable devices such as tablets connected to or even embedded in measurement instruments can share data over the internet in order to achieve the following:

- Measurement and processing synchronization and dynamic interaction.
- On-the-filed accuracy estimation and erroneous observation detection.
- On-the-field access to online shared data both for downloading and uploading measurements.
- Multiple synchronized TSs sharing data in order to speed up the on-field measurement progress and also to collaboratively achieve the detection of critical measurements that are missing.
- On-the-field metadata collection and sharing.
- Visualizations of processed data.
- Real-time progress monitoring and dynamic work reorganization over the net.

The objective of this paper is the introduction and early assessment of a framework that combines all of the above. A data organization to meet the needs for surveying engineering applications and accuracy requirements will be proposed, to

facilitate the sharing of VGI information among surveying engineers. Furthermore, TS networking and processing will be described, using data casting technologies and portable processing units along with integrated Web-GIS services. The proposed framework and referred techniques can support new methodologies for land surveying that can largely benefit from applying the above concepts. We introduce the term "Collaborative Cloud Land Surveying" (CCLS) as a method that combines on-the-field measurements, processing, sharing and validation in real-time. The core of the proposed approach is found in the VGI behavior concept for geo-data sharing and exchange.

This paper is structured in 6 Sections. In Sect. 2 (after the current introductory section) the proposed framework is described, analyzing applied techniques and introducing new concepts that can be applied on-the-field. Section 3 discusses an early prototype of a fully functional system developed and used to apply the proposed methodology. As a case study, a large-scale surveying task using networked TS instruments has been executed in mapping of the historic center of Athens; this task is part of the Archaeological Cadastre project of the Greek Ministry of Culture. The results are presented in Sect. 4, analyzing the current implementation's process and accuracy improvements, along with discussion (Sect. 5) and conclusions (Sect. 6) regarding the proposed method's benefits.

2 Method—Architecture

TS manufacturers work towards integrating on-the-field computational tools in surveying equipment. Most of these implementations are limited currently in transformations of coordinate reference systems, as well as in visualizations of points of interest. Surveying Engineers record raw measurements (distance, angles), corrected by the necessary geometric reductions, in order to be able to provide high accuracy position estimates by applying high accuracy compute algorithms (e.g. least squares). Lately, efforts are made in integrating connected portable devices and TSs in order to upgrade their capabilities with minimum cost (Park et al. 2013).

The proposed method aims to integrate the acquisition and processing of surveying-accuracy data, and also to provide access to shared data captured by other surveying engineers. The synchronization of raw measurements allows for real-time data flows from and to any connected TS, while project overview and progress indicators are also available to authorized clients. There are two main types of actors: *Data collector* that refers to all types of activities that capture measurement data on-the-field, and *data manager*, that authorizes users to process collected data. After discussing these entities, a database schema for storing all data is presented; finally the main on-the-field functions are reviewed. Figure 1 illustrates an overview of the proposed architecture. The system components are further analyzed in the following paragraphs.

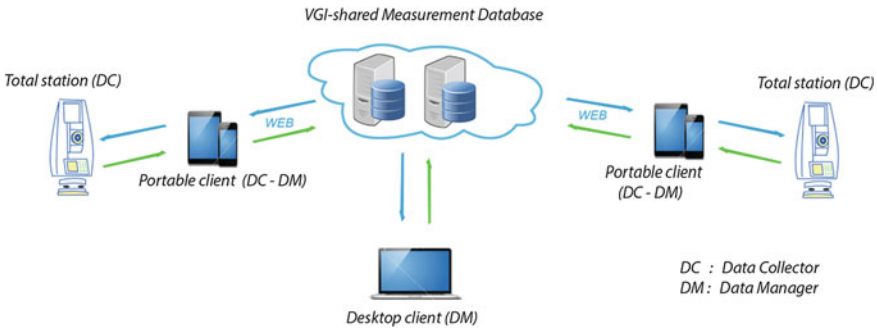


Fig. 1 Networked measurement stations, VGI database and data consumers

2.1 Data Collection

Every device that is used to acquire data on-the-field is referred to as a data collector. The minimum required is TS with data exchange functionality and network access. TSs captures raw measurements of distances and directions. Every single record of data contains at least the following fields: slope distance, horizontal angle, vertical angle and target height. By using as a reference the station position coordinates, along with the above information, the location of any point can be established. Notably, in addition to models that natively support wireless communications, most TSs that allow serial communications for data and command processing can be used together with some aftermarket serial-to-wireless adaptor.

TSs for routine surveying applications do not support networking and visualization functions, nor do they offer any programming framework. On the other hand, powerful handheld portable devices provide processing abilities at very low cost, especially since the introduction of the Android ecosystem. Therefore, any Android tablet or Smartphone doubles as a great tool for data management. In the case study to be discussed in the sequel, a Nexus 10 tablet (10″ screen, 2-core 1.7 GHz CPU, 2 GB RAM) and a LG G2 mobile phone (5.2″ screen, 4-core 2.2 GHz, 2 GB RAM) have been used, connected via Bluetooth to a TS. As far as data collection is concerned, the software that has been developed uses the Bluetooth connection to send the appropriate commands and waits for measurement data to be received back [slope distance (sd), horizontal angle (hz), vertical angle (vz)]. This way, the software takes over the handling of the measurements. The TS receives the commands and responds by supplying the measurement data (angles hz, vz and slope distances sd) as seen in Fig. 2.

Additionally, other portable units can be configured to capture attributes of objects, metadata, tagged photos and further manage network data flows. Due to the fact that even TSs with limited programming abilities need to be used, as already mentioned, the portable devices become the middle layer for routing data to

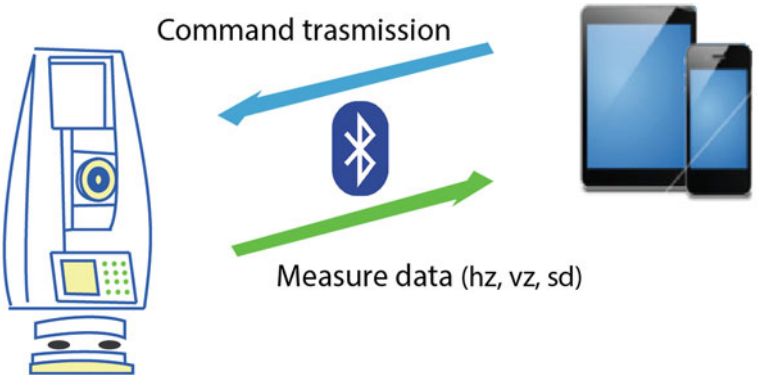


Fig. 2 Portable device total station communication

a geo-database, via a mobile data network. This way it is possible to take typical low-cost TSs that only have a common serial interface, and transform them into networked devices.

2.2 Data Modeling

Geographic information is meant to be gathered in a web-accessible database as surveying engineers upload measurement data, according to the proposed method. The design and development of a database model for this purpose is of essential importance to support real world applications. Also the data model should overcome the information heterogeneity issue (Elwood 2008), using a global modeling approach over collected data. In the sequel a first database model that has been developed and used, will be discussed. Figure 3 shows the entity—relationship diagram as it was integrated in the prototype software developed for this project.

Control Points (CP) are points on the earth's surface whose location has been accurately defined and can be identified on ground, buildings, structures etc. They form networks (reference networks) and are used as the infrastructure for computing all other points' positions, making their use of critical importance. Routinely surveying applications usually require that such a network is already established. However, lack of access to past data while on-the-field, makes the established CPs useless. The goal of our approach is to make the established CPs reusable, which means that anybody can have access to their data. A CP is defined by a description, location, accuracy, time, creator and a global id. There is not always a priori knowledge of the coordinates, but in cases of projects related to datum and national reference system (such as the archaeological cadastre project described in Sect. 3) it is essential to use known CP coordinates.

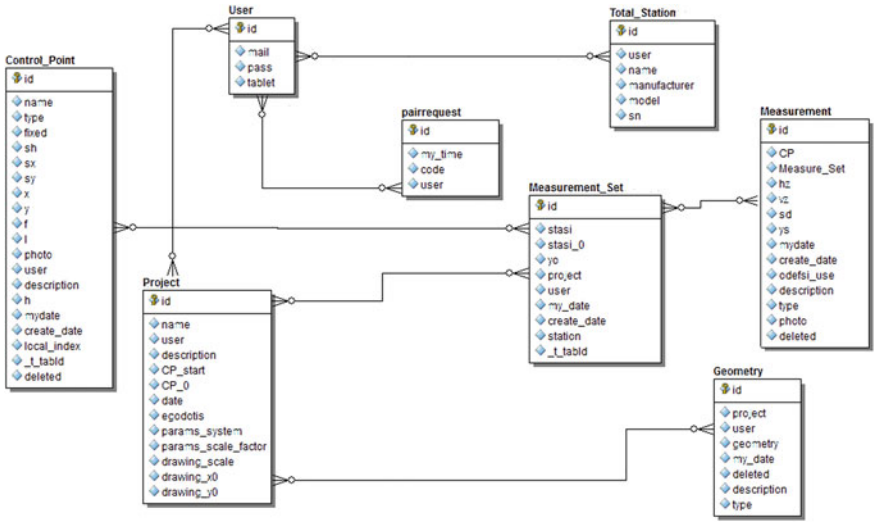


Fig. 3 Entity—relationship diagram

Every time a TS is set over a CP in order to measure, there is a set of measurements that is recorded. The raw measurements are grouped into sets of data, as they share similar properties defining "Measurement_Set" model object. The Measurement itself is the core entity. As the raw data come from the TS, an instance of a measurement class is created. Basic attributes include the horizontal and vertical angles, slope distance, target height and meta-data. In order to enable high accuracy post processing methods that combine existing and collected data, raw measurements accompanied by their associated "measure of quality" are the objects that get uploaded to the system repository. Additionally, raw measurements coming from multiple users, are processed by a quality assessment function that rates combinations of users and instruments; ratings can then be used as filtering criteria by other surveying teams. The computed coordinates are stored in a global reference system (Earth-centered earth-fixed (ECEF or ECF) coordinates) while at the same time users can select the preferable project datum to apply the required computations, transformations, etc.

The above objects are the minimum needed to define the model. Additionally, timestamps and other relevant metadata that refer to spatial resources (Danko 2012) could be used to define an ontology-based approach (Fonseca et al. 2002; Ramos et al. 2013) to describe each point; notable the current implementation does not support this, which will be considered in a future update. Modeling, encoding and computing over spatio-temporal data, allow multi-vocality of measurements over space and time. As the position of measured features could change over time (e.g. sidewalk reconstruction, building movements after earthquakes, infrastructure network reform), the proposed approach allows for temporal management of measurement to track phenomena of such nature. As stated, the use of shared CPs

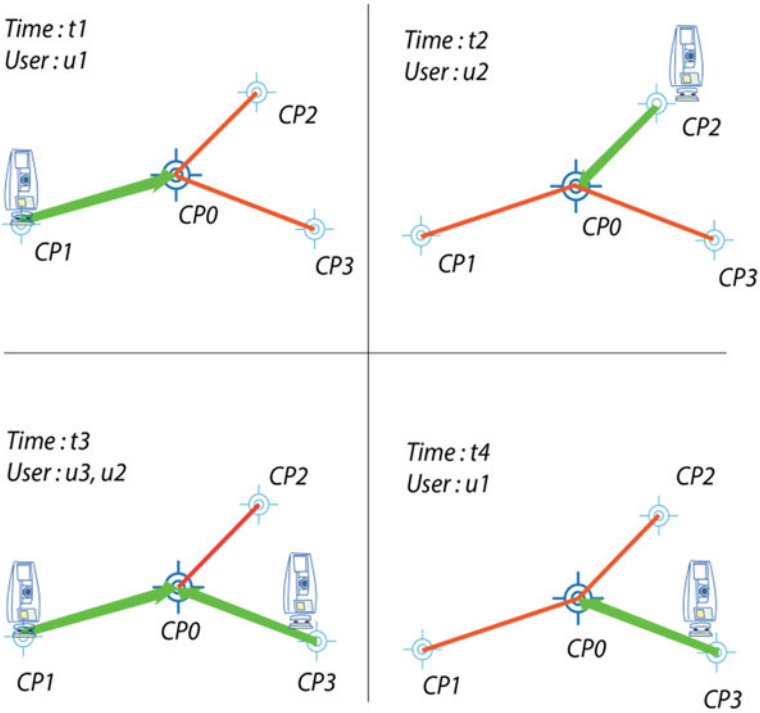


Fig. 4 Multi user—time position CP definition

would be beneficial, as these are modeled objects and multiple measurements over these would increase coordinates precision. The same principle can apply to any network of interest. Figure 4 describes different cases of determining the position of the same Control Point (CP0). There are approaches like multi spatial (when CP0 is determined by different CPs), multi user (when different users determine the position of CP0) and multi epoch (when CP0 position is determined over different timeframes). This allows the determination of the accuracy of user equipment as well as for the detection of time-based changes.

2.3 Data Management—Process

Data management and processing are both procedures that must be executed in real-time, to allow access to all available information on-the-field as well as in the office. This approach examines two types of system clients, namely portable (on-the-field) and desktop (office). Each client type receives and offers distinct functionalities using appropriate tools and functions, which will be discussed, in the following sections.

2.3.1 Portable Client

The portable devices, as described in Sect. 2.1, interact with the TS, in order to capture raw measurement data. Together with the data collection, the devices are employed for three more important tasks: data routing, data processing and information visualization. Details about each task are given in the following sections. Appropriate software for this project has been developed in Android OS that enables all the above operations to be executed.

Portable client—data routing The portable clients perform the data routing, since TSs have limited functionality. The first step is the control of the TS over Bluetooth, which is followed by the measurement data response. The developed software gathers the raw measurement data which may be enriched with other types of data (e.g. photos, metadata, spatial attributes) essential to extent geometry and enrich potential usability (Poore and Wolf 2013; Mohsen 2013) this data is stored locally in order to have offline access, and is also sent to the system server over a wireless Internet connection. The final goal is to achieve data synchronization both on user request and in real time when possible.

Portable client—data process One of the main advantages of the proposed architecture is the real-time data processing during the data collection on-the-field. This allows the surveyor to validate the collected measurements, detect erroneous observations, verify the integrity of measurements by eliminating a possible lack of measurements—as the real-time processing can detect missing information, and integrate all available data. In order to make this possible, computations of the reference network are triggered to compute the positions of the entire CP network upon any new measurement data entry. This way, whenever the local device or any connected network device provides new data, the network CPs positions are updated so that the user can constantly evaluate the full dataset. Doing so, makes it possible to detect erroneous observations, as conflicting measurements get highlighted, prompting for a review.

Portable client—data visualization Portable devices are equipped with high definition flat panel displays capable of providing an advanced visualization experience. The developed software displays both raster maps and vector generated data. Geo-referenced maps, web map service (WMS)—tiles and orthophotos of the area of interest are pre-loaded on the device and be used as a background of overlaid vector data. In the project described in this work, orthophotos provided by the National Greek Cadastre Service are used as a background, providing 20 cm accuracy level over urban areas, allowing for gross error detection—removal (every measurement that contributes in over 20 cm position error of measured point, is immediately recognized). Regarding vector information, there are multiple cases of spatial data usage. Preloaded vector files can be projected over the project workspace, in order to be compared with the collected data (*.kml files have been used for our case study). Every time the system recalculates a feature's position, it gets drawn over the raster images and the available layers containing the vector

information. As mentioned above, this results in the detection of erroneous observations, which are highlighted on the screen. Figures 5 and 6 show visualization modes—User Interface (UI) tools as developed and used in the current implementation.

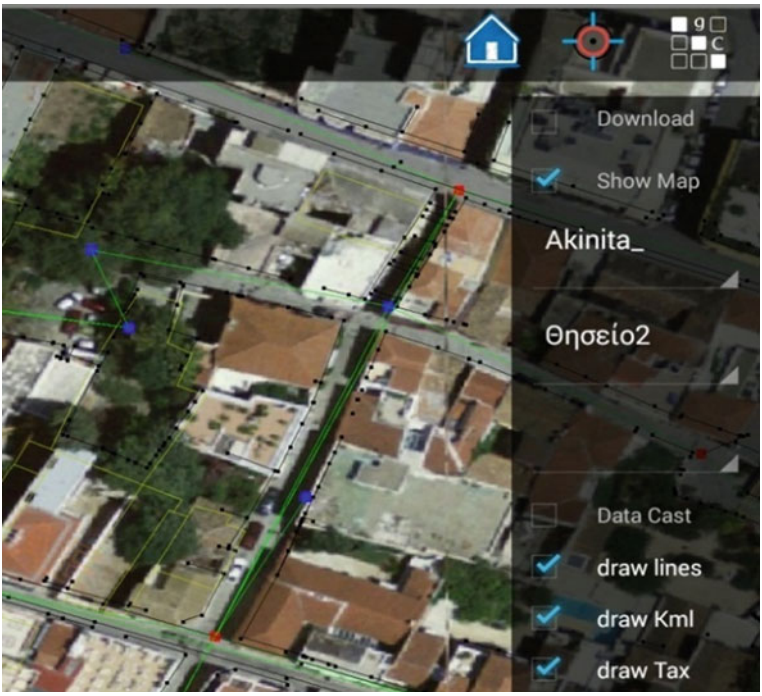


Fig. 5 Portable client WMS visualization

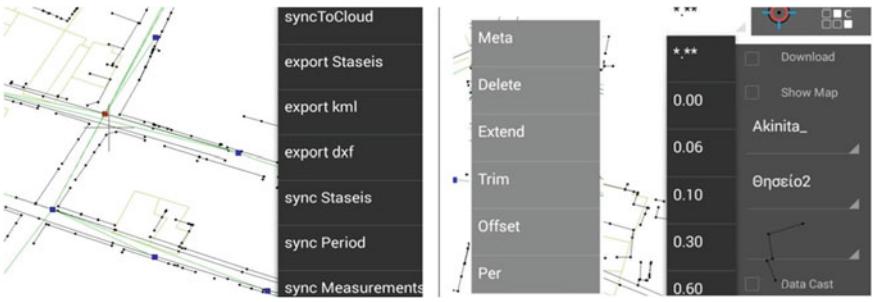


Fig. 6 Portable client vector visualization

2.3.2 Desktop Client

The Project administration—overview (including the field work monitoring), is also possible via dedicated software. The Desktop client developed for this project runs on a web browser environment, enabling the project management and granting administrative rights (project creation, global variable setting, grant user access, set available layers, etc.). Also there are several functions provided additionally to those of the portable clients, such the project creation, project edit, progress overview, computations finalization, report export and quantitative tools for the purposes of this research. Moreover, as the web application has been developed in JavaScript, HTML, and PHP programming languages, it is possible to hand out the system functionality through an application program interface (API), which will allow further extensions by the community, according to the current trend in platform-independent collaborative software development.

2.4 Client-Server Software

Centralized data management requires a database for maintaining data. Through the selected Database Management System (DBMS), the developed software can implement data management functions such as input, storage and retrieval, while ensuring both data integrity and security. The current implementation uses MySQL which is an open software DBMS, on a typical Linux distribution (Ubuntu), which has all the advantages of Open Source, such as low-cost, vendor-independence and extensibility.

The client software, on the other side, is developed over Android 4, using the Java Eclipse IDE. The portable devices are able to offer the appropriate functionality deriving from modified calculation algorithms to match the project needs. The software for this project has been developed in Android OS, which is an open OS, and also is the most popular platform of the majority of tablet devices. Research on Android usage on-the-field has been already in progress (Park et al. 2013).

3 Case Study

In order to test and validate the functionality of the proposed approach, it was used in a real project of the Greek Ministry of Culture; this project is about the mapping of the historic center of Athens including all the archaeological sites, monuments and the private real estate property, as part of the Archaeological Cadastre. As a result, mapping of the area should provide spatial information of places of interest. The study area is about 460,000 m^2, 60 % of which is urban area of high density.

This project is ideal for the proposed system because is a large-scale application giving the opportunity to collect and manage large amount of measuring data

Fig. 7 Boundary of project area over OpenStreetMap and satellite image

coming from multiple work groups at the same time. Figure 7 visualizes the boundary of work area over OSM and satellite image.

An important aspect within the project's scope is the cost class of the equipment. In order to allow as many as possible surveying engineers to participate, the proposed approach requires only TSs in the mid-cost range. The case study described, was based on a low priced Kolida KTS-442RC TS (angle accuracy 2″, distance accuracy [±(5 mm + 2 ppm · D), non prism, ±(2 mm + 2 ppm · D), prism]). The medium to low end TSs currently do not support wireless data transfer in their vast majority apart from RS-232 communication. In order to allow TSs for routinely surveying applications to be used, a Bluetooth to serial adaptor can be integrated to enable wireless data transmission and command execution.

Field work used the android application as data collector and data manager. Bluetooth adapter were used to establish connections between TSs and tablets. Tablets manage commands to the TSs, as well as data synchronization. Also, all computations needed for the measured data to be visualized in real-time and the accuracy to be also calculated in real-time. Multiple data collectors/TSs collected the project data that was processed and displayed simultaneously by all clients. The above schema is illustrated in Fig. 1.

During a 3-month data collection period, 8 surveyor engineers and several archaeologists worked together in mixed teams. Participants had from 0 to 20 years of working experience; three groups were measuring with TSs on-the-field. To understand the usability of the proposed approach, some groups used proposed system during the measurement process, and some others worked on field using the classical surveying workflow. As a result, it was possible to make a comparison regarding productivity and accuracy boost of this approach, as described in Sect. 4. At an initial level, the approximate point position for each property and archeological monument, was located using the existing address along with the Google maps search service, so that the field work would have approximate reference points (Fig. 8a, b). Furthermore, datasets for some properties were available containing non validated information (such as older topographic maps). Finally web mapping

Fig. 8 Approximate position of points of interest over **a** OSM and **b** satellite image

service (WMS) of the Greek National Cadastre and Mapping Agency S.A. provided background maps of 20 cm accuracy used to overlay both existing and measured data.

Up to October 2014, the reference network consisted of 224 CPs covering about 50 % of the total project area. After filtering out inaccurate data, 6974 measurements that were acquired on the field have been used. The CPs and reference network density are shown on Figs. 9 and 10, respectively.

Fig. 9 Control points over satellite image

Fig. 10 Reference network over satellite image

Figure 11 depicts part of the created geometries as the Desktop Client overlays on an OSM map. The user interface (UI) allows to select applicable backgrounds (OSM, Cadastre WMS) while thematic layers can be turned on and off by check-boxes on the main bar. There are also multi type CPs that are shown as points with different colors and sizes in order to be able to distinguish property type on site. The sidebar on the left is used to view data of selected properties and set attributes (e.g. state, description, and other info). Finally images taken on-the-field can be uploaded and viewed through the current interface.

Fig. 11 Created geometries as desktop client overlays on OSM map

4 Results

As described in Sect. 3, measurements have been acquired applying both classical and proposed surveying methodology. After the final computations of the collected measurements from the reference network, a comparison of the results accuracy was made between those produced by the proposed approach and those obtained by the classical surveying practice. Traverses (branches of reference network, consisting of several CPs) were also structured having CPs measured using both the typical and the proposed approach (CCLS). In the process of traverse solution, the angular and linear error is estimated by comparing measurements to known geometric information. The angular error is defined as the divergence between measured angles and known geometries (e.g. a measured square would sum to 359° 59′ 40″ indicating an error of 0° 0′ 60″ to perfect geometry), and linear error as the divergence between computed and known point coordinates. The above errors get distributed to each CP. Table 1 gives the error information of traverses measured in the field, so that accuracy evaluation of proposed methodology and comparison to classical surveying is possible. Columns 2 and 3 give angular and linear solution error respectively of each traverse. Columns 5, 6, and 7 refer to measurement method of CPs, while columns 8–11 distribute the total error to the two different methods used. For example, record 5 analyzes traverse S58-33-116, consisting of 10 CPs, 6 of which were measured using classical surveying and 4 using proposed system. Traverse angular error was computed to 0.0178° (0.0107 for classical surveying and 0.0071 for CCLS) while linear error was 0.102 m (0.061 m for classical surveying and 0.041 m for CCLS). After the division of sum of errors by number of CPs respectively, both average angular and linear error per CP for each method is given. These results show that by following the CCLS approach, the angular error has been reduced by 60 % while the linear error has been reduced by 25 %.

Additionally, another interesting result is the productivity change. For the same work time on the field, there were 64 CPs needed to be set by the teams following the classical surveying approach, while only 53 CPs required by those who followed the proposed method. Given the fact that the field groups that followed the classical surveying approach consisted of three members, while on the other hand only two were needed for the proposed method, it can be deduced that there is a cost/productivity benefit of the proposed method.

Given the fact that the teams that participated in this project had no previous experience in applying the proposed method, the productivity and accuracy are expected to improve even further. Completing the project should provide more data to analyze in order to get more feedback.

At this point, some benefits that came up through the process should be noted. After the final computations, during the production of the 2D plans, there were several cases where some surveying engineers had to revisit the field in order to confirm the dimensions or other missing information. None of these cases had used the proposed method, which indicates further the effectiveness of the approach.

Table 1 Measurement computations—error estimation (angular units −degrees × 10^{-3}, linear units—mm)

1 Traverse	Error		4 Traverse length (m)	Control points			Angular error		Linear error	
	2 ang (deg 10^{-3})	3 Linear (mm)		5 Total	6 Surveying	7 CCLS	8 surv	9 CCLS	10 surv	11 CCLS
S41-S47	70	18	207.47	5	1	4	14	56	4	14
S44-S46	0.7	9	74.59	2	2	0	0.7	0	9	0
S48-S60	17.8	75	118.97	4	4	0	17.8	0	75	0
S58-116	1.1	99	725.54	15	3	12	0.2	0.9	20	79
S58-116	17.8	102	311.14	10	6	4	10.7	7.1	61	41
S58-116	1.1	85	99.59	16	0	16	0	1.1	0	85
S73-S86	14.3	61	99.59	3	3	0	14.3	0	61	0
S78-S68	19.6	100	170.93	5	5	0	19.6	0	100	0
S80-S93	24.6	9	212.52	7	7	0	24.6	0	9	0
S101-S106	17.9	69	134.51	4	4	0	17.9	0	69	0
S112-ST65	24.9	55	316.03	18	18	0	24.9	0	55	0
SG10-SG67	15.4	79	257.8	4	0	4	0	15.4	0	79
SG11-SG24	16.1	17	308.22	8	0	8	0	16.1	0	17
SG13-SG56	5.8	21	175.72	5	0	5	0	5.8	0	21
ST65-S131	23.3	29	141.13	11	11	0	23.3	0	29	0
Total				117	64	53	155.5	51.9	492	336
Average							2.4	0.98	8	6
Error reduction								0.6		0.25

Moreover, during the field data collection, there were cases where more than 2 CPs had been set within few cm spacing by different users over time, making difficult to determine the correct one. These cases are considered as error sources, so users had to measure all CPs, in order to be sure not to miss the correct one. Afterwards, during the post-processing procedure, each of these CPs had to be used separately in the solution in order to detect which one is the correct. Alternatively, groups that followed the proposed approach were automatically notified of the measurement and the respective solution error. Table 2 summarizes the differences between classical surveying and CCLS; Table 3 shows more advantages of the proposed approach.

Table 2 Key differences

Point	Surveying	CCLS
Work flow	2 step flow, data collect (field) data process (office)	Data collect—process unification
Total station topology	Isolated working node	Part of an interactive network
Data form	Distance-angle station depended, data files	Structured database modeling spatial information along with metadata
Time frame	Static, time fixed object description	Multi epoch data collection, temporal measurement repository
Project overview review	Time dependent incoherent project overview after every data collect—data process cycle	Real time project progress—overview, continuous remote review
Data flow	Field data collection saved to local media	Real time data route from and to CCLS database
Data access—reusability	Limited access—availability, hard to integrate due to lack of modeling	Real time open access through web service, easy to integrate—structured information

Table 3 Potential benefits

Property	Benefit	Description
Data recycle	Cost reduction	Use existing data, speed completion time
Field process	–Accuracy improvement	Continuous comparison to existing data, real time model solution
	–Detect erroneous observations	
	–Spatio-tempo feature tracking	
	–Large scale multiple station approach	
	–Interactive network ontology oriented data approach	
	–Direct availability	

5 Discussion

This paper has presented findings from an early experimentation with a methodology that challenges the classical topographic surveying process by using VGI along with modern collaborative network-based concepts. Though VGI projects have been usually fed by data that citizens provide, it was initially defined as the harnessing of tools to create, assemble, and disseminate geographic data provided voluntarily individuals. In this manner, individuals are not discriminated as professional users or not. Communities of users with high expertise would greatly benefit from such a practice, while at the same time non-experts would also have access to high quality data. Surveying engineers will both provide and request raw data, using ordinary measurement equipment and a mobile smart device, but also citizens will greatly benefit from products that integrate this kind of available information.

The proposed method introduces novelties in the way data are collected and processed, unifying both these processes. It introduces field networking for TSs while a central data store is used to synchronize all the connected devices that now have access to the full dataset that is available while on the field.

The case study presented has applied the proposed method and conclusions were drawn for both the system operation in the field and the processing as well as the significant raise of productivity. The results indicate a substantial error reduction by 60 % on angular measurements, while the linear error reduction is estimated at 25 %. Ensuring data quality and credibility is of critical importance in such an approach, as VGI related research has points (Bhaduri and Devillers 2012). Additionally, a productivity raise of 24 % during the corresponding measuring period has been achieved, regarding both the quality and quantity of collected data.

Considering that all the information will be stored in an online repository, allowing reusability by authorized users, the dataset is expected to grow rapidly. This kind of data feed creates self-expanding and continuously self-improving networks, like reference networks, power stations, hydrographic networks etc. Common VGI data coming from citizens without appropriate knowledge have not yet proven to meet the standards of topographic base projects (David 2013). By using the proposed approach, the area of "Social Surveying Engineering" (a term defining scientific behavior of sharing raw surveying measurement data by specialized users) can be expanded thus enabling the development of VGI projects of special interest and high accuracy demands, allowing for the first time the re-use of large-scale spatial information of Engineering-level accuracy.

Production cost should decrease by both productivity raise and equipment upgrade. The application developed for this project, has been set on android OS and requires merely a TS that has a basic serial interface that accepts terminal commands. This transforms a low budget, high accuracy equipment, to a networked device accessing multisource—multi type data instrument with up-to-date processing power and abilities which can improve the surveying methodology.

6 Conclusions

This work sets a new framework for land surveying, integrating volunteer geographic information that users provide through appropriate services. Current technological achievements allow the creation of a system that would provide such functionalities, while at the same time data networks allow information sharing in real time. Benefits of this new concept have been analyzed and results show that accuracy and productivity both raise remarkably.

There are many open questions regarding issues such as dataset development—sharing—usage evolution in this specific scientific area. Such architectures that would enable geographic information integration are currently under research (Pinto 2000). Globally, interest is focused on community-created, yet quality-evaluated content that offers multiple benefits. Surveying engineering evolves this way, as recent trends have proven to be enabling new approaches.

Future work will integrate the full dataset as soon as measurements are available for the whole area of interest. Updated results shall complete this stage of evaluation and provide further comparisons regarding accuracy and productivity. Moreover, future projects that integrate currently collected information will allow over time reusability and enable spatiotemporal data processing, revealing the potential of geographic information sharing among surveying engineering community members.

Acknowledgments Measurements and data produced in a study undertaken by the Greek Ministry of Culture about the mapping of the historic center of Athens (part of the Greek Archaeological Cadastre project) have been used in this work.

References

Agrawal, R. (2003). Information sharing across private databases. In *ACM SIGMOD International Conference on Management of Data* (pp. 86–97). New York: ACM ISBN 1-58113-634-X; doi:10.1145/872757.872771.

Bhaduri, B., & Devillers, R. (2012). Role of volunteered geographic information in advancing science: Quality and credibility. s.l., GIScience. p. http://www.ornl.gov/sci/gist/workshops/2012/index.shtml.

Danko, D. M. (2012). Geospatial metadata. *Springer handbook of geographic information* (pp. 191–244). Berlin: Springer.

David, C. (2013). *Potential contributions and challenges of VGI for conventional topographic base-mapping programs.* s.l. The Netherlands: Springer. ISBN Print 978-94-007-4586-5, ISBN Online 978-94-007-4587-2, doi:10.1007/978-94-007-4587-2_14.

Dobson, M. W. (2013). VGI as a compilation tool for navigation map databases, 17. In D. Sui., S. Elwood, & M. Goodchild (Eds.) *Crowdsourcing geographic knowledge* (pp. 307–327). Netherlands: Springer.

Elwood, S. (2008). Geographic information science: New geovisualization technologies—emerging questions and linkages with GIScience research. *Progress in Human Geography*, pp. 1–8. doi: 10.1177/0309132508094076.

Fonseca, F. T. et al. (2002). Using ontologies for integrated geographic information systems. *Transactions in GIS, 6,* 231–257. doi:10.1111/1467-9671.00109.

Ghilani, C. D., & Wolf, P. R. (2008). *Elementary surveying: An introduction to geomatics*. New Jersey: Prentice Hall (ISBN-13 978-0132554343, ISBN-10 0132554348).

Goodchild, M. F. (2011). Citizens as sensors: The world of volunteered geography. *GeoJournal, 69*, 211–221. The Netherlands: Springer. (Print ISSN: 0343-2521, Online ISSN: 1572-9893, doi:10.1007/s10708-007-9111-y).

Moffitt, F. H. (1998). *Surveying*. Boston: Addison-Wesley. ISBN 0673997529, 9780673997524.

Mohsen, K. (2013) Crowdsourced metadata supporting crowdsourced geospatial data. *International Workshop on Action and Interaction in Volunteered Geographic Information* (*ACTIVITY*). Leuven, Belgium: AGILE 2013. http://flrec.ifas.ufl.edu/geomatics/agile2013/programme.html.

Ozyer, T., Kianmehr, K., & Tan, M. (2012). *Recent trends int information reuse and integration.* New York: Springer. (ISBN Print 978–3-7091-0737-9, ISBN Online 978-3-7091-0738-6, doi:10.1007/978-3-7091-0738-6).

Park, J., Lee, S., & Suh, Y. (2013). Development of an android-based App for total station surveying and Korean society of remote sensing. *Korean Journal of Remote Sensing*, *29*(2), pp. 253–261. ISSN Print: 1225-6161, ISSN Online: 2287-9307.

Pinto, J. K. (2000). Information sharing in an interorganizational GIS environment. *Environment and Planning B: Planning and Design*, *27*, 455–474. doi:10.1068/b2652.

Poore, B. S., & Wolf, E. B. (2013). Metadata squared: Enhancing its usability for volunteered geographic information and the GeoWeb. In *Crowdsourcing geographic knowledge* (pp. 43–64). Springer: The Netherlands, http://dx.doi.org/10.1007/978-94-007-4587-2_4.

Ramos, J. M., Vandecasteele, A., & Devillers, R. (2013). *Semantic integration of authoritative and volunteered geographic information* (*VGI*) *using ontologies.* Association of Geographic Information Laboratories for Europe (AGILE).

Rimal, B. P., & Lumb, I. (2009). A taxonomy and survey of cloud computing systems. In *2009 Fifth International Joint Conference on INC, IMS and IDC* (pp. 44–51). Seoul: IEEE 2009. E-ISBN : 978-0-7695-3769-6, Print ISBN: 978-1-4244-5209-5, doi:10.1109/NCM.2009.218.

A Gamification Framework for Volunteered Geographic Information

Roberta Martella, Christian Kray and Eliseo Clementini

Abstract Using crowdsourcing methods to gather large amounts of useful geodata presents many challenges. One challenge is to attract volunteers to participate in crowdsourcing activities. Several studies conclude that to encourage crowdsourcing it is necessary to take into account people's intrinsic motivation (e.g. fun, altruism, ambition). Gamification is a useful approach to promote people's motivation and engagement. The work we report on in this paper tries to give an answer to the question "How to use concepts provided by gamification in order to motivate individuals to participate in crowdsourcing applications in the geospatial context? How to best combine these two worlds?" We designed a gamification framework for VGI processes and applied the framework to an existing application for evaluation purposes. Such a framework is intended for application developers as a guideline to apply principles from gamification to collect user-generated geospatial data.

1 Introduction

Volunteered geographic information (VGI) denotes a term used in GI Science to indicate data gathered via crowdsourcing: it can be defined as the harnessing of tools to create, assemble, and disseminate geographic data provided voluntarily by individuals (Goodchild 2007). VGI as well as crowdsourcing have caught the interest of researchers and commercial enterprises as a method for gathering large amounts of data at a relatively low cost (and in a short time), compared to more traditional data collection methods.

R. Martella (✉) · E. Clementini
University of L'Aquila, L'Aquila, Italy
e-mail: robertamartell88@gmail.com

E. Clementini
e-mail: eliseo.clementini@univaq.it

C. Kray
University of Münster, Münster, Germany
e-mail: c.kray@uni-muenster.de

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_5

A key open challenge in crowdsourcing is how to motivate people to contribute to crowdsourcing efforts. In the context of capturing people's intrinsic motivations, gamification is an approach that has been gaining popularity since 2010; it is defined as the use of game design elements in non-game contexts (Deterding et al. 2011). Gamification can be beneficial in solving the problem of attracting volunteers because, by introducing game elements, people are potentially more likely to want to participate, without the need for extrinsic rewards. By following principles from gamification research and by integrating game elements with application content, regular crowdsourcing activities can become more attractive and engaging.

Since there are no established techniques to integrate gamification into VGI applications, we developed a gamification framework for VGI applications that describes a set of predefined procedures that serve to gamify the VGI process. In order to achieve this goal, we designed a conceptual model that illustrates which game elements can be used in a specific VGI application, and how they can be used. We then evaluated the model by analyzing the effort required by the user of the gamified application.

Furthermore, the theoretical model has been used to gamify a existing mobile app on Android called CampusMapper and implemented as an Android library in order to obtain the CampusMapperGamified app. After that, we carried out an evaluation with CampusMapper users as a proof of concept of the operations introduced after gamification.

The remainder of the paper is organized as follows. In Sect. 2, we present the state of the art in VGI and gamification. In Sect. 3, we introduce our framework. In Sect. 4, we present the case study of gamifying CampusMapper. In Sect. 5, we discuss an evaluation with end-users. Section 6 concludes the paper.

2 Related Work

In this section, we explore VGI and gamification research by analyzing recent literature and comparing different existing services, which use game design elements in geographic data collection processes.

2.1 Open Challenges for VGI

The availability of user-generated geographic data has improved geographic information in several ways. Spatial data created by citizens can be of great value to many parties, including the government and the public sector. The inclusion of spatial data provided by citizens can be a means to augment and update existing geographic databases, particularly when public funds or staffing are not sufficient (Elwood 2008).

OpenStreetMap (OSM)[1] is undoubtedly one of the largest and most famous VGI projects with a growing number of users. Flickr[2] is another example, which provides large amounts of geographic data through photographs shared by subscribers; these are catalogued and indexed by keywords and tags, as well as by the location where a photo was taken.

Fritz et al. (2009) identify two main open challenges in VGI domain. The first challenge is to attract a wide range of volunteers from all over the world who like to get involved in land cover activities. In this regard, Celino (2013) states that to facilitate the collection of high-quality VGI, human sensors need also to be motivated to be helpful and cooperative. Fritz et al. (2009) point out that this first challenge can be solved by games, which make the task of land cover validation more attractive; social networks and existing groups with expertise in geography, however, can be used as further low-cost outreach facilities.

The second challenge is to be able to guarantee a data quality that is appropriate for the task at hand (Qian et al. 2009; Flanagin and Metzger 2008 and others). The amount of information acquired is unpredictable and discontinuous, and its distribution is variable since the data is collected by non-professionals. Consequently, it is not possible to ensure that the collected data is reliable (Qian et al. 2009). These characteristics pose substantial challenges to VGI data management, especially regarding new data cleaning methods.

In this paper, we focus our attention on the first challenge.

2.2 *Participating in Crowdsourcing: Contributors and Their Motivations*

How best to involve people in crowdsourcing projects is not a question that can be easily answered. In this regard, it is important to know who the contributors are, what the reasons are that drive them to participate, and where motivations come from. Regarding the first and second question, (Coleman et al. 2009) classify the types of people who volunteer geospatial information into *Neophyte, Interested Amateur, Expert Amateur, Expert Professional,* and *Expert Authority.* They also identify eleven motivations for user contributions and analyze how many of these motivations apply to VGI applications.

Psychologists have divided users' motivations into two categories: intrinsic and extrinsic. Over the years, many studies have focused on or made use of these motivations. For example, Kaufmann et al. (2011) created a worker motivational framework for paid crowdsourcing environments (such as Amazon Mechanical Turk[3]). The developed model for motivating workers in crowdsourcing

[1] http://openstreetmap.org (accessed 24 Jan 2015).

[2] http://flickr.com (accessed 24 Jan 2015).

[3] http://mturk.com (accessed 24 Jan 2015).

environments focuses on intrinsic and extrinsic motivations. It is broken down into five categories: *enjoyment-based motivations, community-based motivations, immediate payoffs, delayed payoffs,* and *social motivations*.

Eickhoff et al. (2012) hypothesize that there are two major types of workers with fundamentally different motivations for offering their workforce on a crowdsourcing platform:

- *Money-driven workers* are motivated by extrinsic factors like financial rewards.
- *Entertainment-driven* workers are motivated mostly by intrinsic factors without renouncing to financial rewards.

In this paper, we are particularly looking at intrinsic motivation resulting from game-like elements and their combination with VGI.

2.3 Keeping Afloat the User's Intrinsic Motivation: Gamification

Groh (2012) identifies three principles to bring a greater focus to intrinsic motivations as opposed to extrinsic motivations. Initially, such principles were adopted in Self-Determination Theory (SDT) by Ryan and Deci (2000) and introduced by Deterding (2011). Aparicio et al. (2012) identify the same principles as three social and psychological needs to maintain an intrinsic motivation in the user. These are: relatedness, competence and autonomy.

The underlying theory of motivation in SDT also finds increasing acceptance as fruitful approach to the motivational psychology of video games. Indeed, SDT has been shown to integrate many different findings and concepts related to the motivational pull of video games into a small set of constructs embedded in one macro theory of human motivation (Deterding 2011). Furthermore, several empirical studies find strong correlations between video game features, need of satisfaction, and other relevant constructs like enjoyment or intrinsic motivations.

In this regard, Eickhoff et al. (2012) strongly focus on entertainment-driven workers by framing crowdsourcing problems as games. In this way, they increased the degree of satisfaction entertainment-driven workers experience.

Using game elements in order to motivate players to collect geo-data has been analyzed by several studies. In this regard, Celino (2013) notes that gamification of information gathering tasks can be adopted to provide the incentive scheme for VGI. Davidovic et al. (2013) present their work in progress on building a system that uses a location-based game called MapSigns for mapping real world objects as by-products of the game. Crowley et al. (2012) aim to engage users and hope to create a sticky and viral user experience by adding game elements into a social reporting application.

2.4 Cases Explored

To complete the analysis of the state of the art, we analyzed various existing geospatial applications to investigate which game elements they already employ and how such game elements are integrated into them. The applications we considered are: Google Map Maker,[4] WikiMapia,[5] Kort Game,[6] Map Roulette for OpenStreetMap,[7] Urbanopoly (Celino et al. 2012), and Waze.[8]

Google Map Maker is a multilingual service (web mapping site) launched by Google in June 2008, designed to expand the breadth of the service currently offered by Google Maps.

According to its website, "WikiMapia is an open-content collaborative mapping project, aimed at marking all geographical objects in the world and providing a useful description of them." It aims to, "create and maintain a free, complete, multilingual and up-to-date map of the whole world. Wikimapia intends to contain detailed information about every place on Earth."

MapRoulette is a gamified approach to fixing OSM bugs that splits common OSM data problems into micro tasks, while Kort is a WebApp that helps to improve OSM data. Urbanopoly is a mobile location-based game with a purpose (Celino et al. 2012). Waze is the world's largest community-based traffic and navigation app. Table 1 summarizes the main game elements of the applications described above.

3 A Gamification Framework for VGI

From the analysis of literature and applications, we identified the key concepts and main relationships between gamification and VGI. These were the basis for developing the gamification model defined here. In the following, we introduce the definition and main elements of the model, the relations among human needs, game mechanisms and player types, and the relations between VGI elements and game elements. Finally, we suggest the guidelines to use the framework on behalf of application designers.

[4] http://www.google.com/mapmaker (accessed 5 Dec 2014).

[5] http://wikimapia.org (accessed 5 Dec 2014).

[6] http://play.kort.ch (accessed 5 Dec 2014).

[7] http://maproulette.org (accessed 5 Dec 2014).

[8] http://waze.com (accessed 5 Dec 2014).

Table 1 Comparison of different applications with respect to game elements

	Google map maker	Maproulette	Kort game	WikiMapia	Urbanopoly	Waze
Points			yes	yes	yes	yes
Bonus/extra				yes	yes	yes
Leaderboards			yes	yes	yes	yes
Levels				yes		yes
Badges/awards	yes		yes	yes		
Achievements (missions, challenges)	yes	yes	yes	yes	yes	yes
Rating/votes	yes			yes	yes	yes
Avatar						yes
Virtual goods/services ownership					yes	
Special roles/operations	yes			yes		yes

3.1 Model Definition

A high-level representation of the designed model is shown in Fig. 1. The model takes into account the triple "*User, Geo-data, Tasks*". In the gamification process, Users become *Players*, Tasks are encapsulated in *Challenges* and Geographic data can be associated with *Virtual goods*. The type of information that describes the

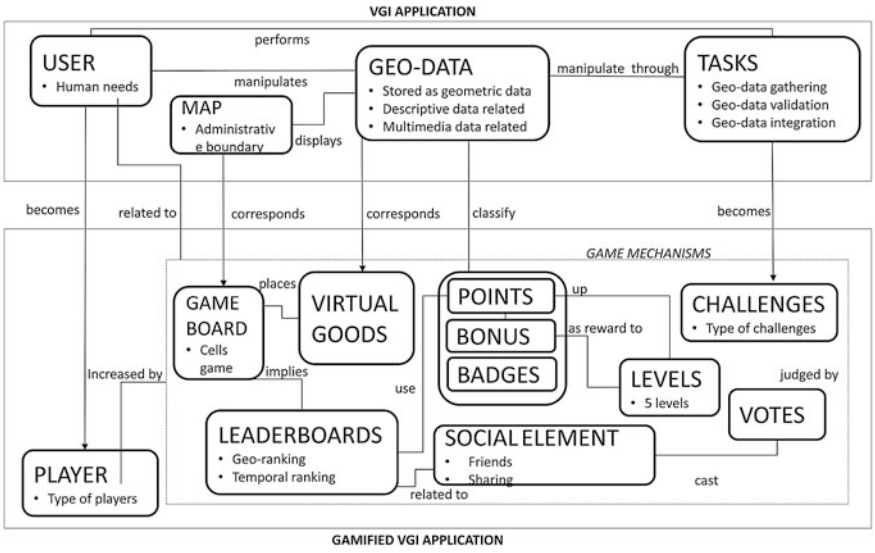


Fig. 1 Conceptual model for the gamification of VGI applications

Geographic data is a parameter used to earn a *Score* and *Badges*, which serve to *Level* up and climb *Leaderboards* (geographical ranking and temporal ranking). Furthermore, the map where geographic data is viewed, is replaced by a *Game board*: existing administrative boundaries can be used as cells to permit players to conquer territories when they are owners of the greater number of virtual goods in that area. The users have to evaluate each performed task to stop malicious actions. The interaction between users takes place, for example, by adding friends and by sharing their own experiences on social networks. All human needs are mapped to game mechanisms as well as the player types.

3.2 Model Main Elements

There are three main elements characterizing any VGI application and making it different from other applications. They are the users, the type of data users manipulate through the execution of the tasks (which is geographical in nature), and the tasks users performs.

The users are volunteers who contribute their time and skills in order to perform what the application requires. Users need to be encouraged in their endeavors. In this regard, their human needs have to be triggered.

Geographic data is a collection of information that can describe objects and things with relation to space: it is stored as geometric data types such as points, lines, and polygons. Usually, it has associated descriptive data such as name, short description, category and/or multimedia data, e.g. pictures. It is visualized on a Map.

Regarding the tasks, we identify three main tasks that the application may permit to perform. They are:

- the Geo-data gathering task, which is the task that includes the insertion of geo-data, modification-deletion of geo-data, insertion of missing information to an element (including georeference);
- the Geo-data validation task, which is the verification of data truthfulness after their creation or updating by leaving a feedback (discussions, comments, judgments, truthfulness level);
- the Geo-data integration task, which is the identification of redundant data and the merging into a single valid data set.

Regarding gamification, we choose game elements such as points, badges, bonus, virtual goods, avatar, leaderboards, levels, friending, ownership, and votes. In addition, as in any game, we identify a player and a game-board. The player, according to Bartle (1996), can be of four types: Achiever, Explorer, Socializer, and Killer.

The choice has fallen on these game elements since it is clear from the literature that they produce good results and are more suited to the crowdsourcing context. Moreover, they appear to be those most frequently used.

3.3 Relations Among Human Needs, Game Mechanisms and Player Types

Game mechanisms indicated in Table 1 are those used by our conceptual model. We perceive them to be the most suitable ones in this type of context (following the analysis of the literature and analysis of applications).

Human needs identified are therefore: Self-expression, Social recognition, Competition, Progression, Reward, Ownership, Cheating, Achievement, and Curiosity. Self-expression can be defined as an exhibiting expression of one's own personality. Avatar and virtual goods can be used to let this dynamic emerge. Social recognition is the need of feeling to be part of a group. The need to interact with others is the stronger need that drives a player (Zichermann and Cunningham 2011) and is important to build a community, encourage competition and collaboration. This need is linked with mechanisms such as friending, sharing experiences, gifting, group quest, etc. Competition is the basis for most of humanity's progress and evolution. With that being said, different personality types have different feelings about competition: sometimes, excessive competition can make players shy away or it can hurt cooperation. The best way to trigger this desire is to see a leaderboard; the desire to climb the standings and to finish first among all is usually very strong. Users have the need to go forward, to make progress, and to see the improvements. Progression can be implemented primarily by the mechanism of levels. Rewards are feedbacks for the work done. They constitute core elements of gamification. Known rewards are points, bonuses, achievement badges or levels, the filling of a progress bar, and providing the user with virtual currency. Ownership relates to the dynamic of "wanting" something. When a player feels ownership, he innately wants to make what he owns better and own even more. That is implemented by Virtual goods that can be everything. Achievement desire instead, can be associated to the completion of challenges and missions. Cheating goes hand in hand with game play if not properly taken into account in the design phase. Before and during game design, it is necessary to imagine how players could possibly cheat or exploit some flaws in the design. However, if anti-cheating measures are overdone, the user experience can be hurt. Curiosity is the need of wanting to find out what actually happens. There are many ways to increase or create curiosity, e.g. using mysterious locked items, treasure chests, and random rewards. Triggering self-expression and social recognition increases the number of Socializer players that are, we remember, the majority of players (Zichermann and Cunningham 2011). Rewards as well as the desire to compete, progress, cheat, or possess lead to an increase of the number of Achiever players. The extensive use of game elements that attract these players, also increases the number of Killers, who constitute only a small minority of players. Explorer players instead are attracted by introducing game mechanisms such as challenges, locked elements, random elements and others.

The relations among human needs, game mechanisms, and player types are recapitulated in Table 2.

Table 2 Relations among human needs, game mechanisms, and player types

Human needs	Game mechanisms	Player types
Self-expression	Avatar, virtual goods	Socializers
Social recognition	Friends, sharing	Socializers
Competition	Leaderboards	Achievers/killers
Progression	Levels	Achievers/killers
Reward	Points, badges, bonus	Achievers/killers
Ownership	Virtual goods	Achievers/killers
Cheating	Voting others	Killers
Achievement	Challenges	Explorers
Curiosity	Random elements, locked elements	Explorers

3.4 Relations Between VGI Elements and Game Elements

Player profiles contain more information than user profiles, such as, avatar, earned rewards (points, badges, virtual goods), rank, and status.

In order to involve a user in performing Tasks, they need to be encapsulated in Challenges. Challenges provide players with a way to brag (indirectly) about what they have done as well as add character to a game. They can be easy, difficult, surprising, or funny. We identify three relevant types of challenges, taking inspiration from Celino et al. (2012): "creative", "quiz", and "rating". Creative challenges mean the challenges draw from a metaphor. Quiz challenges include missing information challenges or multiple-choice questions. Rating challenges ask players to rate aspects according to a given scale, e.g., using values such as "good", "not good", "I don't know". Each challenge type can be subject to a time limit (1 min to complete the task), to a competition (playing against another player at the same challenge), to a conditional lock (challenges locked until player does not complete the previous task), or to completing a combination of things (Combo) to get an achievement. Naturally, all challenges have to be well visible on the Map; the only case that prohibits that a challenge is viewed on a Map is when the location of data is not defined (for example, if the user has to enter geo-data or has to add location information to non-georeferenced data). For non-georeferenced data, data have to be shown to the user in another way.

The Map can correspond to a Game board and the territories' administrative boundaries to cells on the game board. In this way, the attention of users can be captured by the conquest of a territory: from municipalities, to regions, and to an entire country. Ownership of an area can change when a player possesses more geographical elements (virtual goods) than any other player. Users can be attracted also by the fact that they can play in only one part of the territory, e.g., the municipality where they live. Other territories might be locked until something happens that unlocks them. If the VGI application considers a limited territory, for example a city, then a player could be conquering city districts or smaller areas that need not to be districts.

Geographic data can be seen as Virtual goods: a user can collect geo-data in the same way as a player can build virtual goods. Similarly, a user can validate geo-data in the same way as a player can check or perform maintenance of virtual goods. Virtual goods can be owned by the player who contributes most to those goods. If a player spends more time than others on one good, he/she can become its owner. However, he/she has to be careful not to relax too much as otherwise, ownership might be transferred to someone else.

The more users perform tasks, the more points they earn and thus increase their chances of getting a badge. The value of points and types of badges can vary according to several parameters. Generally, there are two types of badges: those with or without milestones (represented, for example, by stars). Badges without milestones do not vary over time. Therefore, once owned, they remain as they are Badges with milestones mean that players can obtain the badge with one star when they succeed in performing the action required to claim that badge for the first time. Further stars can be obtained when further objectives are achieved.

According to Coleman's types of contributors and Raph Koster's Mastery Mountain, we identify five levels: neophyte, interested amateur, expert amateur, expert professional, and expert authority (Coleman et al. 2009). Each level is associated with a range of points, which usually varies according to the breadth and depth of contributions. In fact, the first level is always shorter than the second level because the first one is related to a smaller range and, therefore, easier to overcome.

Regarding leaderboards, there are different types. We propose to adopt a geo-leaderboard. A geo-leaderboard shows the competition among players who live in the same region, contribute in the same area and are in some other way connected to that region. Temporal rankings should also be included, for example, an all-time-best ranking and one referring to a shorter time (e.g., weekly, monthly), in order to give new users a chance to see themselves in a higher ranking.

Data truthfulness is a problem that relates strongly to the correctness of the user. As discussed above, Coleman identifies some motivators for making damaging contributions. We observe that such contributions also come from very competitive users as a result of the introduction of game elements such as points and leaderboards. By entering incorrect information, players can cheat and gain points if this behavior is not stopped by the application We think that if the player is judged by other players in relation to his/her work, then he/she is compelled to operate correctly. Such a judgment can be realized via a vote, either positive or negative, or via a commenting or rating mechanism.

Social elements make us feel part of a group. The need to interact with others is a strong need that drives a player (Zichermann and Cunningham 2011). An application should thus include means to add someone to a list of "friends" (a mechanism called "friending"). This in turn facilitates exchanging messages with each other, watching a friend's leaderboard, and viewing more information about a friend. Further desirable features include the connection to social networks such as Facebook, Twitter, or Google Plus.

3.5 Guidelines for Using the Framework

Based on the discussion above, we have derived a number of guidelines for designers, who intend to use this framework for the gamification of a VGI application.

The first step consists of the identification of the three main elements characterizing any VGI application. Hence, the designer needs to identify:

- who the users are;
- how the geo-data is structured; and
- which tasks users perform that are among those identified by the model.

The second step involves selecting the game elements to integrate into the application (all or a part of them)—starting from those directly connected to the three main elements. In this regard, the designer needs to consider the users' intrinsic motivations in order to increase their participation. It is important to avoid catering to a single player type alone.

In the third step, designers need to adapt and design the application to facilitate gamification. An example about how to apply the guidelines is shown in Sect. 4.1.

4 Case Study: Gamifying the CampusMapper App

This section presents an example application of our framework to an existing VGI application (namely, CampusMapper). CampusMapper is a mobile application for collecting indoor data at University of Münster. With this Android application, users can take photos of public escape plans available at each floor and digitize features such as corridors, rooms, doors, entrances or stairs by drawing them on top of the floor plan image.

4.1 Applying the Framework to the CampusMapper App

Following the guidelines of our framework, as a first step we need to identify the elements of the application, that is, Users, Geo-data, and Tasks. Users are all people that are connected to the University of Münster (e.g., students, professors, trainees). The core task, i.e. the main purpose of the application, is to collect indoor data. This includes the insertion, modification, and deletion of indoor data as well as the insertion of missing information.

In this case, Geo-data can be a corridor, a room, a door, an entrance, or a stair/elevator. Basic shapes have been adopted to facilitate data input in a smartphone touchscreen: we assume that a corridor is digitized as a line, a room as a point (e.g., its center) and described by a name and people inside it, a door is digitized as a line

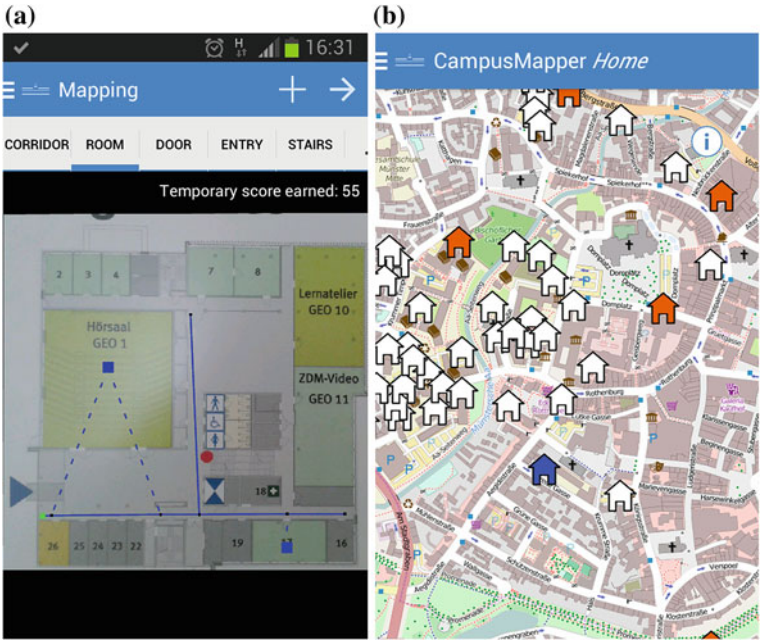


Fig. 2 **a** The basic shapes used to input spatial data in CampusMapper. **b** CampusMapperGamified home screen shows the city of Münster's map and 308 buildings of its University

connecting two rooms, the main entry to the floor is a point, and stairs/elevator are also represented as a point (see Fig. 2a).

As a second step, we decided to integrate in the application rewards such as points and bonus points. Points can be assigned based on the "type of information involved in the task". We assigned to the player an amount of points varying based on the geometric features he/she has introduced: 5 points for each point feature and 10 points for each line. Besides, 2 points for the name of the room, one additional point for each person in the room, and 1 point if the user connects stairs to other stairs. Regarding bonus points, we assigned 10 points to the user who inserted at least each type of feature.

According to the framework, we identified five levels. Points inside levels are distributed considering the number of buildings university and the number of floors of all buildings.

We also integrated some leaderboards: one which shows all players score in the last 15 days, in last month or all time, another one which shows all players score who contributed in some determinate areas. For this latter leaderboard, the definition of the area is fundamental. We identified 2 areas in Münster: the center of Münster and its left area which contains more than an half of all buildings.

Besides, we made possible that a user can conquer a building. The conqueror of a building will be the player who earns most points for that building. Further, to

make the ownership more interesting, we considered also the floor owners, which are the players that got the higher score for those floors. In this way, we subdivided owners in two levels.

4.2 CampusMapper App Gamification Result: CampusMapperGamified App

The result of gamification of CampusMapper is CampusMapperGamified. The home screen shows a map centered on the city of Münster, having superimposed 308 building icons representing all buildings of the University of Münster (Fig. 2b).

Building icons can be white, orange or blue. A building is white when no player has collected data for it. It is blue when the player has the highest number of points on that building (and as a result he/she became its owner). Finally, a building is orange when another player (different from the viewing player) has the highest number of points on it.

A dialog appears when the user clicks on a building. For a white building, a single option appears (only "Collect data"), while for other colors two options are available ("Collect data" and "Look owners") (see Fig. 3a). Name and coordinates of a building are indicated in the title of the dialog. By clicking on "Look owners" for an orange or blue building, a dialog with all owners of the building (first and second level, as described in the previous paragraph) appears. By clicking on "Collect data", the same procedure of data collection as in the regular CampusMapper app would start. There are only two differences: the user no longer has to select a building (because it is already selected), and he/she now earns points when inserting features.

When a user enters a feature during the mapping process, the center of the Mapping screen displays a small popup with the points earned for that feature. Green points are collected by adding features (therefore, he/she gained some points). While red points are gained by removing features (therefore, he/she lost some points). Figure 3b depicts an example for this. In addition, the temporary score that the user has obtained so far is shown at the top of the screen. When the user uploads data, the system checks whether he/she can also receive bonus points. If this is the case, the system updates the temporary score that will be sent to the server together with the image and collected data.

By clicking on the item Leaderboard in the main menu, a tab consisting of two elements appears: Total score and Geo score. Total score displays the rankings according to the scores players earned in the last 15 days, last month and all time. Geo score displays the rankings according to the scores of players, who have earned points in the center and west area of Münster.

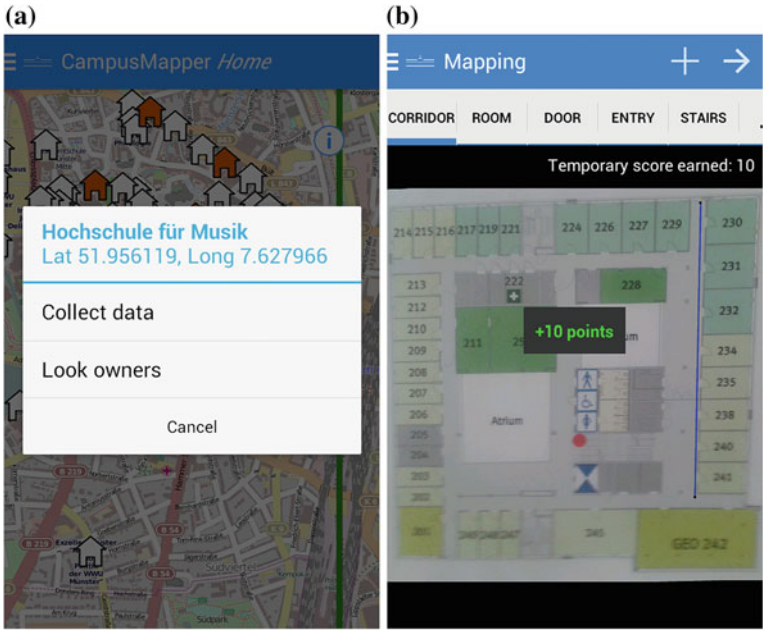


Fig. 3 **a** Dialog appearing after clicking on the "Hochschule für Musik" building (which is an *orange* building) in the CampusMapperGamified App. **b** Adding a corridor increases the score by 10 points. Such an increase is shown by a toast in the CampusMapperGamified App

5 Evaluating CampusMapperGamified

In an initial evaluation, we assessed the effectiveness of the gamification on CampusMapper by involving CampusMapper end-users. The evaluation had a two main goals:

- verifying the applicability of the operations introduced by gamification. For example, we wanted to answer questions such as: "Is it possible for a user to conquer buildings belonging to other users?"; and
- understanding how faithful the mapping is to reality and whether mechanisms introduced as a result of gamification encourage users to cheat.

To set up an experiment with users, we compiled a list of four tasks, which correspond to the actual operations that the user has to perform when playing with the application. We also created two different questionnaires: a first short questionnaire focused on personal information and background with gamified mobile applications, while the second one contained questions about the performed operations.

Participants were first received the first questionnaire, then the tasks list, and lastly, the second questionnaire. The tasks list contained the following items:

- conquer a building that you know and that is not yet owned by someone else (white building);
- conquer a building that you know and that is already owned by someone else (orange building);
- earn one of the top three positions in a leaderboard of your choice;
- reach the second level by getting 300 points.

After completing the first questionnaire, participants were explained the purpose and operation of the game and given a practical demonstration. Twenty users participated in the study. All participants were students of the University of Münster, randomly chosen, in an age group between 20 and 40 years. Each of them used an Android smartphone, on which the CampusMapperGamified app was installed and performed the required tasks. The average time spent by each player to play the game was about 15 min.

Analysis of the questionnaires showed that all users completed task number 1, 5 out of 20 users did not complete tasks number 2 and 3, and 4 out of 20 users did not complete task number 4. Regarding tasks 2 and 3, we noticed that 4 of those 5 people (who did not complete tasks) are part of those who answered "No" to the question "Have you ever used gamified applications where you receive scores, badges or any other kind of reward for your interaction with those applications?". Regarding task number 4, 3 of those 4 people are part of this same category as well. People who have not completed the assigned tasks were thus inexperienced in using games and gamified applications.

Regarding the fidelity of the mapping, the results of the questionnaires suggest that participants claim to produce mappings all in all faithful to reality: only 3 out of 20 cheated in the process of mapping because of the scoring mechanism introduced by gamification. Other users introduced wrong information because they did not know enough about the floor chosen for the mapping or voluntarily entered false information since they assumed it was only a test. It is possible that some participants cheated but did not report this in the final questionnaire.

In a second study, we compared the CampusMapperGamified app to the CampusMapper app. We set up an experiment with 28 participants (14 male and 14 female) to assess the impact of game mechanisms when users are performing indoor mapping tasks.

Every participant was handed an Android Smartphone with CampusMapper and CampusMapperGamified installed. The order in which the two apps were used was counterbalanced to avoid order effects. After having used both apps, participants filled in a questionnaire to measure enjoyment and assess preferences regarding the two apps.

Most of the participants preferred the gamified application over the non-gamified one. Men favored the gamified app more than women. While these findings do not constitute a definitive proof that the game mechanisms implemented actually engage and encourage people to perform indoor mapping tasks more than with

non-gamified version, they do provide some initial evidence in this respect. Due to the short duration of the studies, it remains to be seen for how long the positive effects we observed will persist in the long run.

6 Conclusions

Collecting spatial data is a big challenge for science and industry. VGI promises to contribute towards addressing the challenge with the help of volunteers. One weakness of VGI approaches is the need to motivate participants. Gamification is one promising option to tackle this problem by creating and leveraging intrinsic motivation via game elements. In this paper, we proposed a flexible gamification framework that can be applied to a wide range of VGI projects, and that can thus be useful to all developers who want to gamify applications aimed at collecting spatial data.

The model we developed supports the gamification of such applications. It was evaluated with VGI applications developers and end-users of an Android app called CampusMapper. In the first case, we asked developers to gamify their applications in order to evaluate the framework applicability and ease of use. In the second case, we observed end-users playing with CampusMapperGamified app (CampusMapper gamified version) to evaluate the effectiveness of the application. Both evaluations returned promising results, which provides initial evidence for the proposed approach and its usefulness.

Future work will aim to improve the framework based on the feedback given by developers, and to evaluate the approach in other settings, e.g., with a more significant spatial data collection process or in a long-term deployment study.

Acknowledgments We would like to thank Mijail Juanovich Naranjo Zolotov for performing the user study on comparing the gamified and non-gamified application. We would also like to thank the anonymous reviewers for several insights that helped us improve the paper.

References

Aparicio, A. F., Vela, F. L., Sánchez, J. L., & Montes, J. L. (2012). Analysis and application of gamification. In *Proceedings of the 13th International Conference on Interacción Persona-Ordenador*, *October 2012* (p. 17).

Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD Research*, *1*(1).

Celino, I. (2013). Human computation VGI provenance: Semantic web-based representation and publishing. *IEEE Transactions on Geoscience and Remote Sensing*, *51*(11), 5137.

Celino, I., Cerizza, D., Contessa, S., Corubolo, M., Dell'Aglio, D., Valle, E. D., et al. (2012, September). Urbanopoly—a social and location-based game with a purpose to crowdsource your urban data. In *Privacy, Security, Risk and Trust* (*PASSAT*), *2012 International*

Conference on and 2012 International Conference on Social Computing (*SocialCom*) (pp. 910–913).

Coleman, D. J., Georgiadou, Y., & Labonte, J. (2009). Volunteered geographic information: The nature and motivation of producers. *International Journal of Spatial Data Infrastructures Research, 4*(1), 332–358.

Crowley, D. N., Breslin, J. G., Corcoran, P., & Young, K. (2012). Gamification of citizen sensing through mobile social reporting. In *2012 IEEE International Games Innovation Conference* (*IGIC*) (pp. 1–5).

Davidovic, N., Medvedeva, A., & Stoimenov, L. (2013). *Using location based game MapSigns to motivate VGI data collection related to traffic signs*. International Workshop on Action and Interaction in Volunteered Geographic Information (ACTIVITY).

Deterding, S. (2011). Situated motivational affordances of game elements: A conceptual model. In *Gamification*: *using game design elements in non-gaming contexts*.

Deterding, S., Dixon, D., Nacke, L., & Khaled, R. (2011). From game design elements to gamefulness: Defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (pp. 9–15).

Eickhoff, C., Harris, C. G., de Vries, A. P., & Srinivasan, P. (2012). Quality through flow and immersion: Gamifying crowdsourced relevance assessments. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, *August, 2012* (pp. 871–880).

Elwood, S. (2008). Volunteered geographic information: Key questions, concepts and methods to guide emerging research and practice. *GeoJournal, 72*(3), 133–135.

Flanagin, A. J., & Metzger, M. J. (2008). The credibility of volunteered geographic information. *GeoJournal, 72*(3–4), 137–148.

Fritz, S., McCallum, I., Schill, C., Perger, C., Grillmayer, R., Achard, F., et al. (2009). Geo-Wiki. Org: The use of crowdsourcing to improve global land cover. *Remote Sensing*, *1*(3), 345–354.

Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal, 69*(4), 211–221.

Groh, F. (2012). Gamification: State of the art definition and utilization. In *Proceedings of the 4th Seminar on Research Trends in Media Informatics* (pp. 39–46).

Kaufmann, N., Schulze, T., & Veit, D. (2011). More than fun and money. Worker motivation in crowdsourcing–a study on mechanical turk. In *AMCIS* (Vol. 11).

Qian, X., Di, L., Li, D., Li, P., Shi, L., & Cai, L. (2009). Data cleaning approaches in Web2. 0 VGI application. In *2009 17th International Conference on Geoinformatics* (pp. 1–4).

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist, 55*(1), 68.

Zichermann, G., & Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps.* O'Reilly Media, Inc.

Privacy Preserving Centralized Counting of Moving Objects

Thomas Liebig

Abstract Proliferation of pervasive devices capturing sensible data streams, e.g. mobility records, raise concerns on individual privacy. Even if the data is aggregated at a central server, location data may identify a particular person. Thus, the transmitted data must be guarded against re-identification and an un-trusted server. This paper overcomes limitations of previous works and provides a privacy preserving aggregation framework for distributed data streams. Individual location data is obfuscated to the server and just aggregates of k persons can be processed. This is ensured by use of Pailler's homomorphic encryption framework and Shamir's secret sharing procedure. In result we obtain anonymous unification of the data streams in an un-trusted environment.

Keywords Mobility analysis · Distributed monitoring · Stream data

1 Introduction

Smartphones became a convenient way to communicate and access information. With the integration of GPS sensors mobility mining was pushed forward (Giannotti and Pedreschi 2008). The mobility information of multiple devices is usually stored on a server which performs analysis in order to extract knowledge on the movement behaviour. In the easiest case this is the number of visitors to dedicated places, compare Fig. 1.

The processing of the data streams became infeasible for large use cases, where millions of people are monitored, and massive data streams have to be processed.

T. Liebig (✉)
Artificial Intelligence Unit, TU Dortmund University, Dortmund, Germany
e-mail: thomas.liebig@tu-dortmund.de

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_6

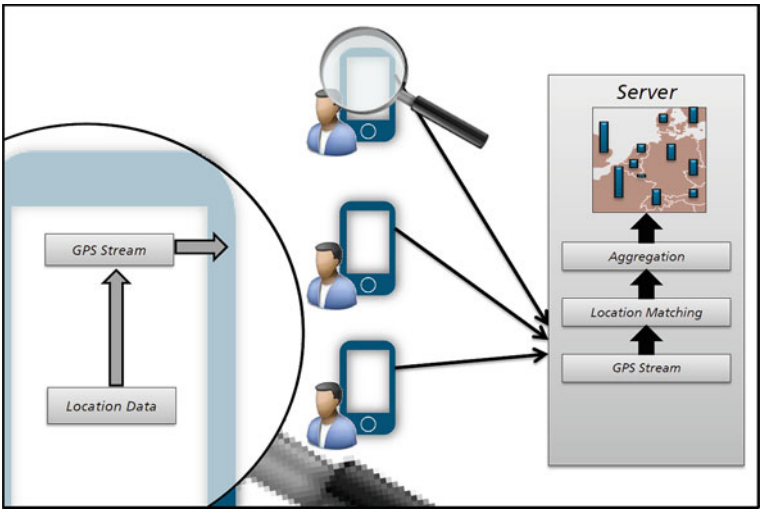


Fig. 1 Centralized mobility data analyis

In this Big Data scenarios, the expensive computation (matching and counting in individual, continuous GPS streams) is split among the parties and just the aggregation step remains in the server in contrast (Boutsis and Kalogeraki 2013) presents a method that distributes the query). Thus, the continuous movement records (GPS) are reduced to episodic movement data (Andrienko et al. 2012) consisting of geo-referenced events and their aggregates: number of people visiting a certain location, number of people moving from one location to another one, and so on. The preprocessing of the GPS data streams is then locally embedded in the location based devices and the aggregation is subject to crowd sourcing. Recent work focusses on in situ analysis to monitor location based events [*visits* (Kopp et al. 2012), *moves* (Hoh et al. 2012)] or even more complex *movement patterns* (Florescu et al. 2012) in GPS streams. In all cases a database with the locations or patterns of interest is provided in advance, and the mobile device computes event-histograms for succeeding time-slices. These histograms are much smaller and may be aggregated by the server in order to achieve knowledge on current movement behaviour, compare Fig. 2.

However, the transmission of these individual movement behaviour still poses privacy risks (Andrienko et al. 2013). Even the access by third parties corrupts individual privacy as recent disclosures on the NSA PRISM program reveal (NSA 2013). The devices monitor daily behaviour and thus reveal working place and hours, the place where we spent the night and other locations indicating information on sensitive subjects as health, religion, political opinions, sexual orientation, etc. Thus, the transferred episodic movement data may even lead to re-identifications.

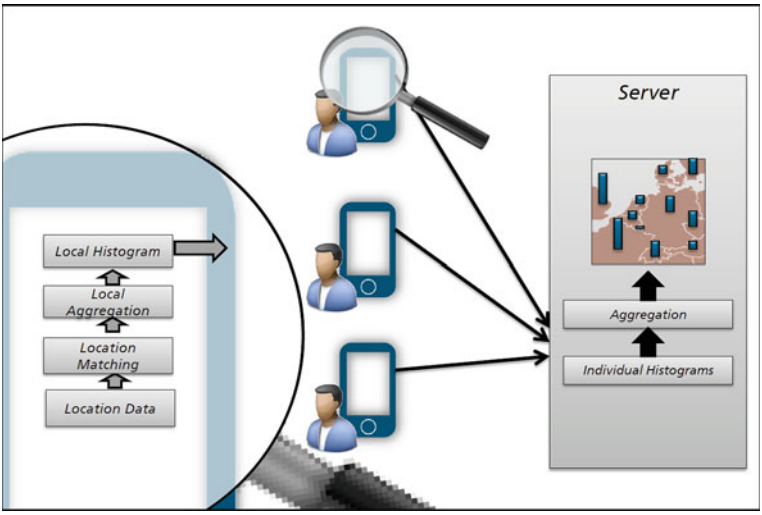


Fig. 2 Aggregation of distributed mobility data streams

The problem we thus focus is the protection of the individual histogram in such a data stream of locally aggregated mobility events. The adversary model is a corrupted server that utilizes the received individual histogram for inferences on the identities and other sensitive data.

Existing methods either act on the network layer (Kopp et al. 2012) or inspired by the differential privacy paradigm they add random noise (Monreale et al. 2013). The work in Clifton et al. (2004) denotes a protocol for secure aggregation among multiple parties, but their algorithm requires extensive communication among the parties and is infeasible in the considered crowd sourcing (i.e. single server) scenario, also their encryption can be broken after several computation cycles.

In contrast, our approach (Liebig 2014) bases on homomorphic crypto systems (Paillier 1999). These are systems where the decryption of several multiplied encrypted values reveals the sum of the original messages. Similarly to the RSA algorithm (Rivest et al. 1983), the system, based on (Damgård and Jurik 2001), uses one-way encryption functions to protect the messages. Thus a public key is used for encryption and a secret private key will be used for decryption. We share the secret key among the clients in the network using Shamir's secret sharing scheme (Shamir 1979). The temporal entanglement of the messages is prevented using a one-way hash as in (Lamport 1981).

The paper proceeds with a detailed discussion of latest work that tackle the described problem. Afterwards our approach is presented in conjunction with preliminaries on crypto systems. However, our approach poses new requirements to the architecture from Fig. 2, which are briefly discussed afterwards. We conclude with a discussion of our achievements and an outlook on future research.

2 Related Work

The problem to protect individual privacy in a distributed scenario with an un-trusted server receives increasing importance with the spread of Big Data architectures and the wide availability of massive mobility data streams. Thus, the problem is subject of many recent publications.

The work in Abul et al. (2008) computes k-anonymity and assumes a trusted server. The work from Kopp et al. (2012) tries to solve the un-trusted server problem by introduction of an obfuscation layer in the network communication, see Fig. 3. But individual location data is identifying, even if it is aggregated in space-time compounds (Monreale et al. 2010). Therefore, this work still delivers the vulnerable data to the server.

Recently, differential privacy was applied to the problem in Monreale et al. (2013). Originated in database theory, differential privacy implies that adding or deleting a single record to a database does not significantly affect the answer to a query (Dwork et al. 2006). The work in Monreale et al. (2013) follows the common method to achieve differential privacy by adding Laplace noise (with the probability density function $Lap(\mu, \lambda) = p(x|\mu, \lambda) = \frac{1}{2\lambda} e^{-|x-\mu|/\lambda}$, where μ is set to zero and $\lambda = 1/\varepsilon$) to every flow value in the vector, as proposed in Dwork et al. (2006), compare Fig. 4.

However, for cell counts differential privacy is known to provide strange behaviour, especially if large number of cells are zero (Muralidhar and Sarathy

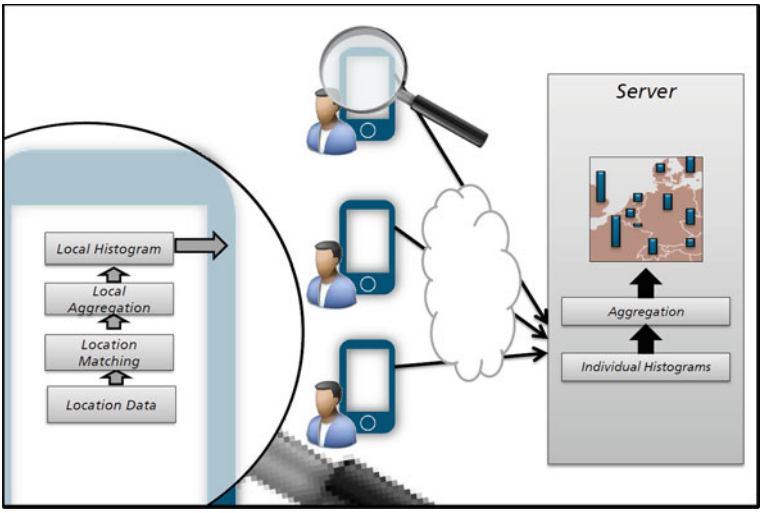


Fig. 3 Obfuscated communication in the distributed monitoring scenario (Kopp et al. 2012)

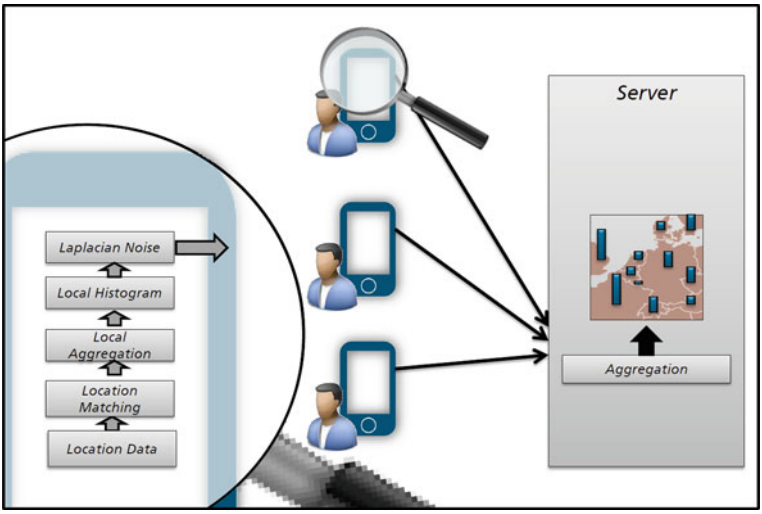


Fig. 4 Differential privacy for the distributed monitoring scenario (Monreale et al. 2013)

2011). Moreover, movement often is a routine behaviour (Liebig et al. 2008) and within their considered time interval most likely similar counts are produced for every person (Liebig et al. 2009), this offers a chance to extract the mean and thus the correct value of the distribution within a stream environment (Duan 2009) as the noise is sampled from $Lap(0, 1/\varepsilon)$ instead of sampling from $Lap(0, m/\varepsilon)$, where m denotes the expected number of queries. Additionally, movement is not random, and thus the frequencies in the vector are not independent, but correlate. Thus, combination of various noisy replies may be utilized to reveal the true distributions.

In contrast, our approach based on homomorphic cryptology in conjunction with a shared key ensures that individual data may not be accessed by the server but only aggregates of at least k people can be used, Since k may equal the number of clients, no data on the individual persons need to be revealed.

3 Proposed Cryptographic Approach

In contrast to previously described approaches our method (1) encrypts the values of the histogram, (2) communicates these ciphertexts to the server, (3) aggregates the ciphertexts and finally (4) decrypts the result, see an overview in Fig. 5. The process utilizes asymmetric cryptography methods using two separate keys: one for encryption and another one for decryption. The utilization of a homomorphic crypto system in conjunction with Shamir's secret sharing guarantees that the individual messages can not be restored, but their sum.

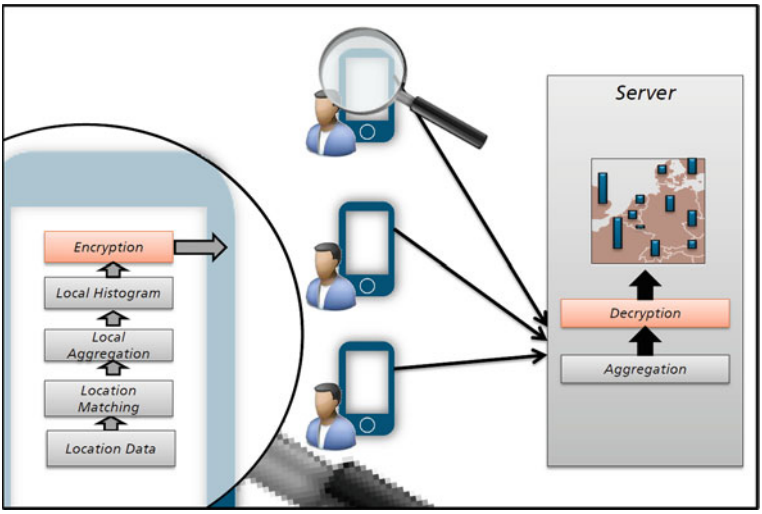


Fig. 5 Proposed privacy preserving aggregation of distributed mobility data streams

As our method bases on the RSA-method (Rivest et al. 1983), homomorphic crypto systems (Paillier 1999; Damgård and Jurik 2001), Shamir's secret sharing (Shamir 1979) and the work on hash chains, described in Lamport (1981), we proceed with a brief primer and describe our method afterwards.

3.1 RSA Algorithm

The RSA-algorithm Rivest et al. (1983) is an asymmetric crypto system. The system bases on two keys, a *private key* which is used for decryption and a *public key* used for encryption. Whilst the public key can be shared with multiple parties, the private key is the secret of the receiver, and may hardly be computed from the public key.

The RSA method uses one-way functions. These are functions which are easy to compute in one direction but difficult to reverse. A simple metaphor of this function is a phone book: While it is easy to derive the call number of a particular person, it is hard to look up the name given a phone number.

Preliminary for understanding is the notion of multiplicative inverse b of a number a, which is defined as $a \cdot b = 1 \mod m$. This inverse just exists, if m and a are co-prime, i.e. $gcd(m, a) = 1$.

Consider a communication among the client who wants to send a message to the server. In this case, the system works as follows. In a key generation process, the server chooses two different primes p and q and computes $n = pq$ and $m = (p-1)(q-1)$. Furthermore, the server chooses a number a which is co-prime to m. The public key, created by the server, then denotes as $pk = (n, a)$. The server computes the multiplicative inverse $b = a^{-1} \mod m$ of a, which is the secret private key.

Encryption:
The client has a message x, with $x < m$. He sends the ciphertext c, computed as

$$E(x, pk) = x^a \mod n. \tag{1}$$

Decryption:
The server decrypts the message and restores the plaintext by computing

$$x = D(c) = c^b \mod n. \tag{2}$$

The system is secure, as knowledge of n does not reveal p and q, since factorization is in NP (Johnson 1984).

3.2 Homomorphic Crypto Systems

A public key encryption scheme (E, D), where E and D are algorithms for encryption and decryption, is homomorphic when it meets the condition $D(E(m_1) \cdot E(m_2)) = m_1 + m_2$ Our approach bases on the generalisation of Paillier's public-key system (Paillier 1999), introduced in Damgård and Jurik (2001). Their crypto system uses computations modulo n^{s+1}, with n being the RSA modulus and s a natural number. By setting $s = 1$ Paillier's scheme is a special case (Paillier 1999). If $n = pq$ with p and q being odd primes, then the multiplicative group $\mathbb{Z}^*_{n^{s+1}}$ is a direct product of $G \times H$, where G is of cyclic order n^s and H is isomorphic to $\mathbb{Z}^*_n$. Thus, $\bar{G} = \mathbb{Z}^*_{n^{s+1}}/H$ is cyclic of order n^s. For an arbitrary element $a \in \mathbb{Z}^*_{n^{s+1}}$ $\bar{a} = aH$ denotes the element represented by a in the factor group $\bar{G}$.

Choose $g \in \mathbb{Z}^*_{n^{s+1}}$ such that $g = (1+n)^j x \mod n^{s+1}$ for known j relatively prime to n and $x \in H$. Let λ be the least common multiplier of $p-1$ and $p-1$, $\lambda := lcm(p-1, q-1)$. Choose d by the Chinese Remainder Theorem, such that $d \mod n \in \mathbb{Z}^*_n$ and $d = 0 \mod \lambda$.

The public key then is n, g whilst the secret key is d.

Encryption:
The plaintext m is element of $\mathbb{Z}_{n^s}$. With a plaintext m we choose at random $r \in \mathbb{Z}^*_{n^{s+1}}$. The ciphertext $E(m,r)$ computes as:

$$E(m,r) = g^m r^{n^s} \mod n^{s+1}. \tag{3}$$

Decryption:
For the ciphertext c compute $c^d \mod n^{s+1}$. If $c = E(m,r)$ this results in

$$\begin{aligned} c^d &= (g^m r^{n^s})^d = E(m,r) \\ &= ((1+n)^{jm} x^i r^{n^s})^d \\ &= (1+n)^{jmd \bmod n^s} (x^m r^{n^s})^{d \bmod \lambda} \\ &= (1+n)^{jmd \bmod n^s}. \end{aligned} \tag{4}$$

In Damgård and Jurik (2001) an algorithm is proposed to compute $jmd \mod n^s$. Their method bases on a function $L(b) = (b-1)/n$ which ensures that

$$L((1+n)^i \mod n^{s+1}) = (i + \binom{i}{2} n + \cdots + \binom{i}{s} n^{s+1}) \mod n^s. \tag{5}$$

The basic idea of their algorithm is to compute the value iteratively in a loop by increasing s, as $L(1+n)^i \mod n^2 = i \mod n$. For convenience, their algorithm is cited in Algorithm 1. With the same method computed for g instead of c the value $jd \mod n^s$ is computed. The plaintext then is:

$$(jmd) \cdot (jd)^{-1} = m \mod n^s. \tag{6}$$

The crypto system is additively homomorphic. As example consider two messages m_1 and m_2 which are encrypted using the same public key pk such that $c_1 = E(s, pk)(m_1, r_1)$ and $c_2 = E(s, pk)(m_2, r_2)$ then $c_1 c_2 = g^{m_1} g^{m_2} r_1^{n^s} r_2^{n^s} = g^{m_1+m_2} r^{n^s}$ so $c_1 c_2 = E(s, pk)(m_1 + m_2, r)$.

3.3 Shamir's Secret Sharing

The work presented in Shamir (1979) discusses how to distribute a secret value d among n parties, such that at least k parties are required for restoring the secret. The idea utilizes a polynomial function $f(x) = \sum_{i=0}^{k-1} a_i x^i$, with $a_0 = d$, and distributes the values $f(i)$ to the parties. In case k of these values are commonly known, the polynomial $f(0)$ can be restored.

Algorithm 1 DAMGARD JURIK ALGORITHM (2001)

```
1:  i:=0
2:  for j := 1 to s do
3:      t1 := L(a mod n^(j+1))
4:      t2 := i
5:      for k := 2 to j do
6:          i := i − 1
7:          t2 := t2 · i mod n^j
8:          t1 := t1 − (t2 · n^(k−1)) / k! mod n^j
9:      end for
10:     i := t1
11: end for
```

The advantage of this method is that the shared parts not larger than the original data. By some deploying strategies of the parts hierarchical encryption protocols are also possible.

3.4 Hash Chain

The work in Lamport (1981) describes a method for authentication with temporally changing password messages. The passwords series are created in advance using a cryptographic hash function which is a one-way function $F(x)$. They are created as follows $F^n(x) = F(F^{n-1}(x))$, where x is a password seed. The passwords are used in reversed order. Thus, the server stores the last value that the client sent, $F^n(x)$, and proves correctness of the new value $F^{n-1}(x)$ by verification of $F^n(x) = F(F^{n-1}(x))$. Afterwards the server stores the latest received value for the next check. As $F(\cdot)$ is a one-way function, the server may not pre-compute next password.

3.5 Putting Things Together

Our cryptographic system follows the protocol of the homomorphic crypto system in Damgård and Jurik (2001). Consider communication among w clients with a single server. Similar to Damgård and Jurik (2001) key generation starts with two primes p and q which are composed as $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes but different from p and q. The RSA modulus n is set to $n = pq$ and $m = p'q'$. With some decision for $s > 0$ the plaintext space becomes $\mathbb{Z}_{n^s}$. Next, d is chosen such that $d = 0 \mod m$ and $d = 1 \mod n^s$. Now, we use Shamir's secret sharing scheme (Shamir 1979) to generate the private key shares of d to be divided among the clients. Thus, we apply the polynomial $f(X) = \sum_{i=0}^{w} a_i X^i \mod l$, by picking a_i

for $(0 < i \leq w)$ as random values from $0, \ldots, l$ and $a_0 = d$, l is a prime with $n^{s+1} < l$. We choose g as $g = n + 1$. The secret share of d for the ith client will be $s_i = f(i)$. A verification key $v_i = v^{\Delta s_i} \mod n^{s+1}$ is associated with each client i. The public key then becomes (n, s, l) and $s_1, \ldots, s_w$ is a set of private key shares.

Encryption:
The plaintext of the ith client m_i', which is element of $\mathbb{Z}_{n^s}$, is multiplied with the one-way hash function $F^n = F(F^{n-1}(a))$ of a commonly known seed a. Thus the plaintext for the encryption results as $m_i := m_i' F^n$. Given this plaintext m_i we choose at random $r \in \mathbb{Z}^*_{n^{s+1}}$. The ciphertext $E(m_i, r)$ computes as:

$$E(m_i, r) = g^{m_i} r^{n^s} \mod n^{s+1}. \tag{7}$$

The client i then communicates $c_i^{2\Delta s_i}$, with $\Delta = l!$ (Damgård and Jurik 2001).

Decryption:
The server can verify that the client raised s_i in the encryption step by testing for $log_{c_i^4}(c_i^2) = log_v(v_i)$. After the required k number of shares S arrived. They can be combined to Damgård and Jurik (2001):

$$\begin{aligned} c' &= \prod_{i \in S} c_i^{2\lambda_{0,i}^S} \mod n^{s+1}, \quad \text{where} \\ \lambda_{0,i}^S &= \Delta \prod_{i' \in S \setminus i} \frac{-i}{i - i'} \in \mathbb{Z}. \end{aligned} \tag{8}$$

Thus, the value of c' has the form $c' = (\prod_{i \in S} c_i)^{4\Delta^2 f(0)} = (\prod_{i \in S} c_i)^{4\Delta^2 d}$. As $4\Delta^2 d = 0 \mod \lambda$ and $4\Delta^2 d = 4\Delta^2 \mod n^s$, $c' = (1 + n)^{4\Delta^2 \sum_{i \in S} m_i} \mod n^{s+1}$. The desired plaintext $\sum_{i \in S} m_i$ can be obtained by previously introduced algorithm and succeeding multiplication with $(4\Delta^2)^{-1} \mod n^s$. The original plaintext can be computed by dividing the resulting sum by F^n. This ensures that previous messages may not be used for analysis of current messages. The homomorphic property of the system is directly used, and bases on the work presented in Damgård and Jurik (2001).

Security:
The security of the crypto system is based on the *decisional composite residuosity assumption* already used by Paillier (1999). The assumption states that given a composite n and an integer z it is hard to decide whether z is a n-residue (i.e. a n-th power) modulo n^2, i.e. whether it exists an y with $z = y^n \mod n^2$.

4 Consequences for the Architecture

As a consequence of our method the keys need to be distributed among the communicating parties: the clients and the server. This may not be done by the server, but has to be performed by a (commonly) trusted authority (TA). Once the keys are distributed, the communication channel to this TA can be closed. Thus, no vulnerable data reaches this third party.

5 Discussion

The hereby presented method overcomes limitations of related work. In addition, our approach may be combined with the methods presented in Monreale et al. (2013). Thus, the transmitted histograms can be obfuscated by Laplacian noise (Monreale et al. 2013). On the other hand transmission may not be obscured by anonymous messages (Kopp et al. 2012) since the identifier of the clients is required for verification of the transmitted messages and reconstruction of the aggregated plaintext.

However, our method assumes that the space covered by individual movements overlaps. If this assumption does not hold, e.g. with persons from different cities, the privacy of each individual is not guaranteed (Abul et al. 2008). An approach to overcome this limitation is by sending messages to the server just if the according entry in the histogram is at least one (i.e. the person was at least once at this location or used at least once the movement pattern). This ensures that the server may just decode the aggregated histogram if a sufficient number of people sent their messages and thus have been there. On the other hand, then the transmission of the message itself contains information on a person's movement behaviour. Thus, future studies should find a message encoding of a zero which does not allow to compute the aggregated sum but passes all verification steps of the server.

Additionally, implementation of this algorithm in heterogeneous distributed environments is essential work in progress in extension of (Schnitzler et al. 2014). The streams-framework (Bockermann and Blom 2012) offers a great platform, as it already contains many methods for data preprocessing and data mining in streams (Bockermann and Blom 2012). As a main advantage it may run standalone on many operating systems (Linux, Windows, iOS) and embedded devices (Android) and also runs on top of the storm-framework.[1]

Acknowledgments This work is funded by the EU FP7 INSIGHT (www.insight-ict.eu) project (Intelligent Synthesis and Real-time Response using Massive Streaming of Heterogeneous Data), 318225.

[1] http://storm-project.net.

References

Abul, O., Bonchi, F., & Nanni, M. (2008). Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (ICDE'08)* (pp. 376–385). Washington, D.C., USA: IEEE Computer Society.

Andrienko, N., Andrienko, G., Stange, H., Liebig, T., & Hecker, D. (2012). Visual analytics for understanding spatial situations from episodic movement data. *KI—Künstliche Intelligenz* (pp. 241–251).

Andrienko, G., Gkoulalas-Divanis, A., Gruteser, M., Kopp, C., Liebig, T., & Rechert, K. (2013). Report from dagstuhl: The liberation of mobile location data and its implications for privacy research. *ACM SIGMOBILE Mobile Computing and Communications Review, 17*(2), 7–18.

Bockermann, C., & Blom, H. (2012). *The streams framework* (p. 12). TU Dortmund University, Technical Report 5.

Bockermann, C., & Blom, H. (2012). Processing data streams with the rapidminer streams-plugin. In *Proceedings of the 3rd RapidMiner Community Meeting and Conference*.

Boutsis, I., & Kalogeraki, V. (2013). Privacy preservation for participatory sensing data. In *2014 IEEE International Conference on Pervasive Computing and Communications* (PerCom) (pp. 103–113).

Clifton, C., et al. (2004). Privacy-preserving data integration and sharing. In *DMKD* (pp. 19–26).

Damgård, I., & Jurik, M. (2001). A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography (PKC'01)* (pp. 119–136). London, UK: Springer.

Duan, Y. (2009). Privacy without noise. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09) (pp. 1517–1520)*. New York, USA: ACM.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography (TCC'06)* (pp. 265–284). Berlin, Heidelberg: Springer.

Florescu, S. C., Mock, M., Körner, C., & May, M. (2012). Efficient mobility pattern detection on mobile devices. In *Proceedings of the ECAI'12 Workshop on Ubiquitous Data Mining* (pp. 23–27).

Giannotti, F., & Pedreschi, D. (2008). *Mobility, data mining and privacy—geographic knowledge discovery*. Berlin: Springer.

Hoh, B., Iwuchukwu, T., Jacobson, Q., Work, D. B., Bayen, A. M., Herring, R., et al. (2012). Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines. *IEEE Transactions on Mobile Computing, 11*(5), 849–864.

Johnson, D. S. (1984). The NP-completeness column: An ongoing guide. *Journal of Algorithms, 5*(3), 433–447.

Kopp, C., Mock, M., & May, M. (2012). Privacy-preserving distributed monitoring of visit quantities. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL'12)* (pp. 438–441). New York, USA: ACM.

Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM, 24*(11), 770–772.

Liebig, T. (2014). Privacy preserving aggregation of distributed mobility data streams. In *Proceedings of the 11th Symposium on Location-Based Services* (pp. 86–99).

Liebig, T., Körner, C., & May, M. (2008). Scalable sparse bayesian network learning for spatial applications. In *IEEE International Conference on Data Mining Workshops, 2008 (ICDMW'08)* (pp. 420–425). IEEE.

Liebig, T., Körner, C., & May, M. (2009). Fast visual trajectory analysis using spatial bayesian networks. In *IEEE International Conference on Data Mining Workshops, 2009 (ICDMW'09)* (pp. 668–673). IEEE.

Monreale, A., Andrienko, G., Andrienko, N., Giannotti, F., Pedreschi, D., Rinzivillo, S., et al. (2010). Movement data anonymity through generalization. *Journal of Transactions on Data Privacy, 3*(2), 91–121.

Monreale, A., Wang, W., Pratesi, F., Rinzivillo, S., Pedreschi, D., Andrienko, G., & Andrienko, N. (2013). Privacy-preserving distributed movement data aggregation. In *Geographic Information Science at the Heart of Europe. Lecture Notes in Geoinformation and Cartography* (pp. 225–245). Berlin: Springer International Publishing.

Muralidhar, K., & Sarathy, R. (2011). Does differential privacy protect terry gross' privacy? In J. Domingo-Ferrer & E. Magkos (Eds.), *Privacy in Statistical Databases* (Vol. 6344, pp. 200–209). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer.

NSA slides explain the PRISM data-collection program. *The Washington Post*. Available: http://www.washingtonpost.com/wp-srv/special/politics/prism-collection-documents/. [Last accessed: 23 June 2013] (06 June 2013).

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT'99)* (pp. 223–238). Berlin, Heidelberg: Springer.

Rivest, R. L., Shamir, A., & Adleman, L. (1983). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 26*(1), 96–99.

Schnitzler, F., Liebig, T., Mannor, S., Souto, G., Bothe, S., & Stange, H. (2014). Heterogeneous stream processing for disaster detection and alarming. In *IEEE International Conference on Big Data* (pp. 914–923). Piscataway: IEEE Press.

Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(22), 612–613.

Part II
Discovering Knowledge and Detecting Changes

Enabling Semantic Search and Knowledge Discovery for ArcGIS Online: A Linked-Data-Driven Approach

Yingjie Hu, Krzysztof Janowicz, Sathya Prasad and Song Gao

Abstract ArcGIS Online is a unified Web portal designed by Environment System Research Institute (ESRI). It contains a rich collection of Web maps, layers, and services contributed by GIS users throughout the world. The metadata about these GIS resources reside in data silos that can be accessed via a Web API. While this is sufficient for simple syntax-based searches, it does not support more advanced queries, e.g., finding maps based on the semantics of the search terms, or performing customized queries that are not pre-designed in the API. In metadata, titles and descriptions are commonly available attributes which provide important information about the content of the GIS resources. However, such data cannot be easily used since they are in the form of unstructured natural language. To address these difficulties, we combine data-driven techniques with theory-driven approaches to enable semantic search and knowledge discovery for ArcGIS Online. We develop an ontology for ArcGIS Online data, convert the metadata into Linked Data, and enrich the metadata by extracting thematic concepts and geographic entities from titles and descriptions. Based on a human participant experiment, we calibrate a linear regression model for semantic search, and demonstrate the flexible queries for knowledge discovery that are not possible in the existing Web API. While this research is based on the ArcGIS Online data, the presented methods can also be applied to other GIS cloud services and data infrastructures.

Keywords Metadata · Semantic search · Linked data · Geoportal · ArcGIS online

Y. Hu (✉) · K. Janowicz · S. Gao
STKO Lab, University of California Santa Barbara, Santa Barbara, CA, USA
e-mail: yingjiehu@umail.ucsb.edu

K. Janowicz
e-mail: jano@ucsb.edu

S. Gao
e-mail: sgao@uamil.ucsb.edu

S. Prasad
Applications Prototype Lab, ESRI Inc, Redlands, CA, USA
e-mail: sprasad@esri.com

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_7

1 Introduction and Motivation

ArcGIS Online[1] is a geoportal developed by Environment System Research Institute (ESRI). It allows GIS users throughout the world to create, edit, and share geo-data, Web maps, services, and GIS tools (Dangermond 2009). To remain manageable, the plethora of ArcGIS Online resources (called *items*) are accompanied by a rich set of metadata, including titles, descriptions, and information about users, user groups, and so forth. Based on these metadata, one can browse through the collection of GIS resources, or sort them by features such as the popularity or date.

Currently, the data and metadata reside in data silos, and can be accessed via a RESTful Web API. However, only queries which satisfy pre-designed templates can be submitted to retrieve data. This hinders flexible knowledge discovery. For instance, if one wants to find out *"which users have produced highly rated maps about natural disasters in the USA"*, such a query has to be first hard-coded into the current API before it can be used. While it is possible to embed a small number of frequent queries, a GIS user can easily come up with a new customized search that has not been designed before. This limitation demands a solution that allows flexible and customized queries.

Meanwhile, as new GIS resources are being created every day, the existing keyword-based search is becoming increasingly limited for finding results that match a user's interests. For example, a search of *natural disasters in Oklahoma* would not be able to return maps about *tornados in Moore*, since the term *tornado* is not in the query and the system does not understand *Moore* is a city in *Oklahoma*. Thus, it is necessary to establish an intelligent search method that can retrieve maps based on the semantic and geographic meaning of the input query.

To enable semantic search as well as flexible knowledge discovery, ArcGIS Online resources should be annotated with machine readable terms which can characterize the map content. Titles and descriptions in the metadata can deliver important information to humans, but they cannot be directly used by machines. While ArcGIS Online also allows users to assign structured tags to maps, those tags are often incomplete or misleading due to the voluntary nature of the data.

To address these restrictions and thus improve the usability of Online GIS cloud services, three steps need to be taken: (I) the metadata provided by the users have to be enriched with machine readable terms; (II) all metadata have to be converted into a format which frees it from data silos and allows flexible queries; (III) a new user interface has to be developed to provide semantic search and enable interesting knowledge discovery. **The contributions of our work are as follows**:

- We present a workflow to enrich the original metadata with machine-readable concepts and named entities.
- We develop an ontology for ArcGIS Online and convert a sample of ArcGIS online metadata into Linked Data.

[1] http://www.arcgis.com.

- We design a semantic search function by expanding input queries and tuning a linear regression model.
- We discuss two flexible queries enabled by our solution, and show the knowledge that can be discovered from the Linked metadata.
- We implement a prototypical Linked-Data-driven Web portal for ArcGIS Online using the presented methods.

This work makes use of the Semantic Web technology stack, including the concepts (Berners-Lee et al. 2001), Linked Data principles (Bizer et al. 2009), Resource Description Framework (RDF) (Hitzler et al. 2011), and other techniques. Such technology stack has been used in existing works to facilitate knowledge discovery (Hu et al. 2013; Keßler et al. 2012). For a more detailed rationale on the use of Linked Data and semantics in GIScience, readers are recommended to (Janowicz et al. 2012). While we have used ArcGIS Online data in this research, the presented methods could also be generalized to other GIS cloud services.

The remainder of this paper is organized as follows. Section 2 provides a brief description on ArcGIS Online. Section 3 discusses the workflow to extract metadata from the API, convert them into RDF, and enrich them with machine-readable terms. Section 4 presents a semantic search method which retrieves GIS resources based on semantic and geographic relevance. Section 5 employs two customized queries to demonstrate the flexible search enabled by the Linked-Data-driven solution. Section 6 describes the prototype implemented as a proof-of-concept. Section 7 summarizes this work and discusses future directions.

2 ArcGIS Online—A GIS Cloud Service

As a collaborative platform, ArcGIS Online enables GIS users throughout the world to create, edit, and share maps, services, and other GIS resources. ArcGIS Online contains a large variety of resources, including Web maps (consisting of a basemap and several layers), services (e.g., map service, feature service, geoprocessing service), as well as document-based data (e.g., shapefiles, CSV files). ArcGIS Online also contains a large number of registered users and user groups, e.g., a *transportation group*. Finally ArcGIS Online also provides a data sharing and reuse mechanism: users can integrate existing services into the maps instead of having to upload all data.

While there are datasets contributed by U.S. Geological Survey (USGS), Federal Emergency Management Agency (FEMA), and other authoritative institutes, a large proportion of the Web maps are volunteered geographic information (VGI). Similar to other VGI, (meta)data quality is one important issue that needs to be addressed (Goodchild and Glennon 2010). In this work, we mainly focus on enriching the metadata of Web maps and services, since it is directly related to the semantic search function which will be discussed later.

The metadata of Web maps and services are recorded in a semi-automatic manner. Information items, such as the map ID, creation date, and the creator's name, are generated by the system automatically, while the creator needs to manually type in a title, a short description (called *snippet*), and several tags. ArcGIS Online uses these tags as annotations to find maps according to a particular topic. The examples below show the titles, descriptions, and tags of three ArcGIS Online maps. While some of the tags are descriptive (e.g., *Thompsons Lake* and *tornadoes*), others are more difficult to interpret or even misleading. For example, the second map is tagged with *book*, while the map is actually about floods. While the tags of the first and the third map are more comprehensive, *landscape* could be one additional tag for the first map to characterize the type of *change*. Similarly, *natural disaster* could form an additional tag for the third map. Due to the voluntary nature, we cannot require users to provide a list that contain every possibly related tag, nor can we mandate the usage of certain pre-defined tags. However, map titles and descriptions often provide useful information about the content of a map, and therefore can be used to extract meaningful tags.

1. **Map title**: Landscape Change: Thompsons Lake, NY
   Snippet (**Description**): Minor landscape changes near Thompsons Lake in the Helderberg's of upstate New York
   Tags: Thompsons Lake, NY, Change, GIS
   URL: www.arcgis.com/home/item.html?id=849ae63adc2c446f9ba54c10a50fbd7b
2. **Map title**: Tragedy and Kindness: Brisbane Floods, January 2011
   Snippet (**Description**): This map shows pictures in Brisbane, Australia in the aftermath of the floods that occurred in January 2011
   Tags: book
   URL: www.arcgis.com/home/item.html?id=07845c87cd7e4f2eb2292b978267b6af
3. **Map title**: Moore, Oklahoma—Tornadoes from the 1950's to the 2000's-Copy
   Snippet (**Description**): Map showing tornadoes in Moore, Oklahoma from the 1950's to the 2000's decade by decade and classified by strength
   Tags: tornadoes, Fujita Scale, Tornado Alley
   URL: www.arcgis.com/home/item.html?id=45f31bea7f624766bf23827ec488d9a3

3 Data Conversion, Ontology Design, and Enrichment

In this section, we describe the process of converting a sample of the ArcGIS Online metadata into RDF and enriching the data with thematic terms and geographic entities extracted from map titles and descriptions.

3.1 ArcGIS Online Data Sample

ArcGIS Online data can be accessed and retrieved using the ArcGIS Online REST API.[2] In this work, we use a sample retrieved between 7 January 2013 and 9 January 2013. This sample contains information about 35,624 Web maps, 13,649 feature services, 5565 map services, 8582 Web mapping applications, 20,725 users, and 2052 user groups. These data can be divided into three categories: *GIS resources* (including maps, services, tools, and so forth), *ArcGIS Online users*, and their *user groups*. For our Linked Data conversion we are especially interested in the relations between those categories, such as: users create GIS resources; multiple users can belong to the same group; a single user can belong to multiple groups; if a user belongs to a group, then her public GIS resources also belongs to this group, and so forth. Following the ArcGIS Online terminology, we will refer to GIS resources as *items*.

3.2 Ontology for ArcGIS Online

An ontology formally restricts the interpretation of domain vocabulary towards their intended meaning, and can be considered as the backbone for data organization. A growing number of well-defined ontologies exist and have been used in many projects, e.g., Dublin Core (dc)[3] and Friend Of A Friend (foaf).[4] Reusing existing ontologies is generally a good practice to facilitate data exchange and integration (Heath and Bizer 2011). However, ArcGIS Online already has an established schema embedded in many of its existing functionalities. While it is possible to semantically align parts of the ArcGIS Online schema to existing ontologies (e.g., from *arcgis*:*owner* to *dc*:*creator*), such a translation may nevertheless bring compatibility issues that may require code revisions in other ArcGIS Online modules. Even more, as (to our best knowledge) there are no existing ontologies for Online GIS cloud services, we would have to import a wide variety of existing ontologies which often leads to unintended logical consequences (Janowicz et al. 2012). Therefore, we design a specific ontology for ArcGIS Online, which can be generalized to other GIS cloud services and aligned to existing ontologies (instead of importing them).

Figure 1 illustrates the major classes and relations of the developed ontology. *arcgis*:*Item* is a general class for all GIS resources, such as Web maps and map services. The particular type of the GIS resource is defined by the class *arcgis*:*Item-Type* whose instances include *arcgis:Web-Map*, *arcgis*:*Map-Service*, and *arcgis*:*Feature-Service*. If an *item* is a Web map, it also has links to its basemap and other

[2] http://resources.arcgis.com/en/help/arcgis-rest-api/index.html.

[3] http://dublincore.org/documents/dcmi-terms/.

[4] http://xmlns.com/foaf/spec/.

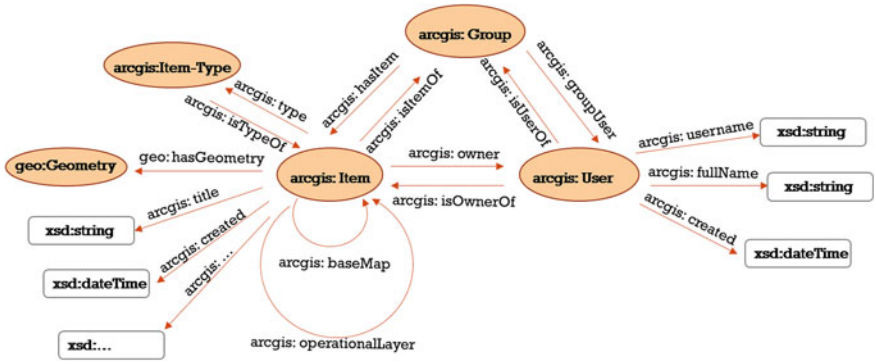


Fig. 1 Ontology for ArcGIS online (ellipses are classes, rectangles are literals)

layers through the relations of *arcgis*:*baseMap* and *arcgis*:*operationalLayer*. The geographic extent of an *arcgis*:*item* is represented using the class *geo*:*Geometry* from OGC's GeoSPARQL vocabulary,[5] which is defined to enable geographic queries based on SPARQL.[6] Figure 1 also shows the interactions among the classes *arcgis*:*Item*, *arcgis*:*User*, and *arcgis*:*Group*. For lack of space, we cannot discuss any axioms in detail here but refer the interested reader to our full implementation using the Web Ontology Language (OWL) at http://sejp.geog.ucsb.edu/esri/ontology.

3.3 Entity Naming

To publish and share high quality data on the Semantic Web, the naming of the entities (e.g., maps, users, and groups) should follow the established Linked Data principles (Heath and Bizer 2011; Keßler et al. 2012). While ArcGIS Online uses a hash string to identify its Web maps, it also provides globally unique HTTP URLs for users to access these GIS resources. Following Linked Data principles 1 and 2, we reuse these HTTP URLs to name the entities in the ArcGIS Online data. Below are some examples for the entity naming.

- **Web Map**: www.arcgis.com/sharing/rest/content/items/be9b7b9fb3514757ba5e6000aa4bd5ba
- **Feature Service**: services1.arcgis.com/10Nf6qqrwDJLkcL2/arcgis/rest/services/Rivers/FeatureServer

[5] http://schemas.opengis.net/geosparql/1.0/geosparql_vocab_all.rdf.

[6] SPARQL (http://www.w3.org/TR/sparql11-overview/) is the query language for graphed data, e.g., Linked Data, standardized by the World Wide Web Consortium (W3C).

- **Map Service**: tiles.arcgis.com/tiles/XWaQZrOGjgrsZ6Cu/arcgis/rest/services/CambridgeBasemap/MapServer
- **ArcGIS Online user**: www.arcgis.com/sharing/rest/community/users/ezgis76
- **ArcGIS Online group**: www.arcgis.com/sharing/rest/community/groups/a707bf7643cf47b89548d0a0184b6950

All of these entity names can be *dereferenced* (by appending "?f = json" to specify the output format), which leads to information about these GIS resources, users and groups. This practice follows the 3rd rule of the Linked Data principles: *published data resources should be self-descriptive*. Currently, we are also working on establishing external links from ArcGIS Online maps to *GeoNames* and *DBpedia* which will satisfy the 4th rule.

3.4 Enriching Data with Geographic Entities and Thematic Terms

Among the rich ArcGIS Online metadata, titles and descriptions often convey useful information about the map content. For example, given a map titled "Los Angeles population density", one can grasp the general idea of the map without having to look into the map. Titles and descriptions are represented in the form of natural language, which is easy for humans to read, but difficult for machines to process.

Therefore, our goal is to extract meaningful terms from titles and descriptions to characterize the content of maps. In contrast to plain text documents, the content of a map can often be divided into two parts: the thematic part and the geographic part. Examples in our sample dataset include maps entitled *"New York Population Density"*, *"California Fires"*, and *"Hurricans in Florida"*, to name but a few. Consequently, our extraction and enrichment process differentiates thematic and geographic terms. This differentiation is important for the functionality of semantic search, as thematic similarity and geographic similarity need to be treated separately (Jones et al. 2001).

We use two Linked-Data-driven and semantically-enabled Web services to extract and differentiate thematic and geographic terms: *DBpedia Spotlight* (Mendes et al. 2011) and *OpenCalais* (Butuc 2009). *DBpedia Spotlight* is an automatic annotation system that can label out the terms that have corresponding entries in *DBpedia* (Auer et al. 2007; Bizer et al. 2009). An important feature of *DBpedia Spotlight* is its capability to disambiguate a term that has multiple matching entries based on the term's context. For example, the term *Santa Barbara* can refer to a

place[7] but also a TV series.[8] To find the most likely *DBpedia resource* for *Santa Barbara*, DBpedia Spotlight applies the TF-IDF (term frequency- inverse document frequency) similarity matching between the surrounding text (i.e., the map titles or descriptions) and the corresponding *DBpedia* content and then ranks the resources according to the matching score. Such disambiguation is possible as DBpedia uses rich ontology, and therefore places and TV series can be distinguished by their types.

Similar to DBpedia Spotlight, *OpenCalais* can extract and semantically categorize entities. While typically Spotlight is able to extract most of the important thematic concepts and geographic entities for our sample data, *OpenCalais* complements those results with *broader terms*. For instance, it will extract *natural disaster*, if *earthquake* is present in a map's title or description. Thus, we employ both services for the enrichment process. Additionally, we also differentiate between the entities extracted from the map titles and those extracted from the descriptions. The list below shows the three ArcGIS Online examples (discussed in Sect. 2) with previous and newly added tags.

1. **Map title**: Landscape Change: Thompsons Lake, NY
   Previous Tags: Thompsons Lake, NY, Change, GIS
   After Enrichment:

   Title thematic terms: change, lake, landscape
   Title geo-terms: Thompson
   Descriptions thematic terms: lake, landscape, minor, Thompsons lake
   Descriptions geo-terms: New York, Thompson, upstate new york

2. **Map title**: Tragedy and Kindness: Brisbane Floods, January 2011
   Previous Tags: book
   After Enrichment:

   Title thematic terms: flood, January, kind, natural disaster, tragedy
   Title geo-terms: Brisbane
   Descriptions thematic terms: aftermath, flood, January, natural disaster, picture
   Descriptions geo-terms: Australia, Brisbane, Brisbane,_Australia

3. **Map title**: Moore, Oklahoma—Tornadoes from the 1950's to the 2000's-Copy
   Previous Tags: Tornadoes, Fujita Scale, Tornado Alley
   After Enrichment:

   Title thematic terms: 1950, 2000, natural disaster, Tornado
   Title geo-terms: Moore, Moore,_Oklahoma, Oklahoma
   Des. thematic terms: 1950, 2000, classified, decade, natural disaster, strength, Tornado
   Descriptions geo-terms: Moore, Moore,_Oklahoma, Oklahoma

[7] http://live.dbpedia.org/page/Santa_Barbara,_California.
[8] http://live.dbpedia.org/page/Santa_Barbara_(TV_series).

Finally, based on the developed ontology, the naming schema, and the data enrichment process, we convert the ArcGIS Online sample to RDF triples using a customized script developed on top of the Jena API[9] and store the Linked Data in the GeoSPARQL-enabled Parliament triple store (Battle and Kolas 2012). The newly extracted terms are inserted into Parliament as triples, and are linked to the corresponding maps to complete the enrichment process. It is worth to note that the original tags voluntarily contributed by users are no longer used due to their varied completeness and potential errors.

4 Semantic Search for Maps

In this section, we discuss our approach to enabling semantic search based on the enriched metadata. Semantic search attempts to understand the meaning of the user's input query, thereby improving the search results (Guha et al. 2003; Zhou et al. 2007). This differs from traditional keyword search which is based on the occurrence of syntactic matches (Tran et al. 2007).

4.1 Query Expansion

The first step towards semantic search is to expand the natural language query from the user to cover related terms. Similar to the data enrichment process, we use *DBpedia Spotlight* and *OpenCalais* to dynamically extract thematic terms and geographic entities, which provide the basic terms for query expansion. The expansion of thematic terms and geographic entities should be treated differently. For the thematic terms, it is important to identify the terms which have similar meaning but different syntaxes, whereas for the geographic entities, place hierarchies and spatial proximity should be taken into account (Jones et al. 2001).

For the expansion of the thematic aspect, we use the UMBC Semantic Similarity Service (Han et al. 2013), which is based on a combination of Latent Semantic Analysis (LSA) (Landauer and Dumais 1997) and knowledge from WordNet (Miller 1995). Given a thematic term, the UMBC Service can find the top n semantically similar terms based on a similarity score ranking. This allows us to also find maps about *reservoirs* if a user searches for *lakes*. For the expansion of geographic entities, we use the GeoNames gazetteer service to find the top 10 nearby and related places. Thus, if the user's query contains *California*, popular places related to California, such as *San Fransisco* and *Los Angeles*, will also be returned as related entities. The list below shows an example for expanding a user's query.

[9] https://jena.apache.org/.

- **User query**: Vacations in Hawaii
  Extracted Terms:

  Thematic term: Vacation
  Geo term: Hawaii
  Expanded Terms:

  Thematic terms: holiday, honeymoon, leisure, picnic, getaway, sabbatical, spring break, camping, leave, resort
  Geo terms: Honolulu, Hawaii County

4.2 Constructing Matching Features

Based on the expanded queries and the enriched map data, we construct matching features to quantify the relevance between a query and the candidate maps. 8 matching features have been constructed using the thematic concepts and geographic entities extracted from titles and descriptions. To avoid mismatches due to minor syntax variations (e.g., *library* and *libraries*, or *California* and *California*), we make use of the stemming technique, and convert terms to lowercase. Detailed explanation for each feature is listed as below:

- **Title Thematic Exact matching** (**TTE**): the number of matches between the original user input thematic terms and the thematic terms in the titles of candidate maps (e.g., *vacation* from the query and *vacation* from the map title).
- **Title Thematic Similar matching** (**TTS**): the number of matches between the expanded thematic terms from the user's query and the thematic terms from the titles of candidate maps (e.g., the term *holiday* expanded from the input term *vacation* and *holiday* in the map title).
- **Title Geographic Exact matching** (**TGE**): the number of matches between the geographic entities from the original user input query and the geographic entities from the titles of candidate maps (e.g., *California* from the input query and *California* from the map title).
- **Title Geographic Similar matching** (**TGS**): the number of matches between the expanded geographic entities and the geographic entities from the titles of candidate maps (e.g., *Los Angeles* expanded from the input term *California* and *Los Angeles* in the title).
- **Description Thematic Exact matching** (**DTE**): the number of matches between the original input thematic terms and the thematic terms in the descriptions of candidate maps (e.g., *water body* from the query and *water body* in the map description).
- **Description Thematic Similar matching** (**DTS**): the number of matches between the expanded thematic terms and the thematic terms in the descriptions of candidate maps (e.g., *lake* expanded from the input term *water body* and *lake* in the map description).

- **Description Geographic Exact matching** (**DGE**): the number of matches between the geographic entities from the original input and the geographic entities in the descriptions of candidate maps (e.g., *California* from input query and *California* in the map description).
- **Description Geographic Similar matching** (**DGS**): the number of matches between the expanded geographic entities and those in the descriptions of candidate maps (e.g., *Los Angeles* expanded from *California* in the input query and *Los Angeles* in the description).

In addition, an interaction variable, namely **Thematic-Geo Interaction** (TGI), has been introduced, which is defined as the sum of thematic matching scores multiplying the sum of geographic matching scores; see Eq. 1.

$$TGI = (TTE + TTS + DTE + DTS) \times (TGE + TGS + DGE + DGS) \quad (1)$$

As denoted by the name, TGI captures the interactions between thematic matches and geographic matches. TGI will have a positive value only when both thematic and geographic matches exist; otherwise it will be zero.

The rationale for introducing this 9th feature is that a good result for map search often needs to have both thematic and geographic matches. Consider a query for *Drugs and Crime in California*. A map that has a high number of thematic matches (and thus a high thematic matching score) but is about *Drugs and Crime in Spain* may not be of interest to the user. On the contrary, a map that has only one thematic match, e.g., *drugs*, but also the geographic match with *California* is more likely to be considered as a good match for a user's query. In fact, we will test this assumption in the evaluation section.

4.3 Ranking Results Using a Linear Regression Model

While we have constructed 9 matching features, a method is necessary to combine these matching features and quantify the relevances between an input query and the candidate maps. Specifically, such a method should satisfy two criteria: (1) correctly rank the relevance between a query and a candidate map; (2) can be easily embedded into a SPARQL query (since RDF has been employed to interlink the data).

We propose to use a regression model to combine the 9 matching features. Such a model can satisfy the above criteria: it can provide fair ranking and can be easily integrated into a SPARQL query. Equation 2 shows the regression model.

$$\begin{aligned} R(q,m) = {} & \lambda_1 TTE + \lambda_2 TTS + \lambda_3 DTE + \lambda_4 DTS + \lambda_5 TGE \\ & + \lambda_6 TGS + \lambda_7 DGE + \lambda_8 DGS + \lambda_9 TGI \end{aligned} \quad (2)$$

where $R(q,m)$ represents the ranking score between query q and map m. $TTE, TTS, \ldots, TGI$ are the 9 matching features respectively, and $\lambda_1, \lambda_2, \ldots, \lambda_9$

are the coefficients for each matching feature. It is worth to note that we deliberately design this regression model without a constant term. This is because when no match exists, $R(q, m)$ should be 0. Therefore, the constant term can be considered as 0, and we force the model to pass through the origin of the axis.

To estimate the coefficients, we designed an experiment with test queries. In this experiment, 7 participants were invited to evaluate the search results for 10 different queries. For each query, we provide a search phrase (e.g., "California population density") and 10 maps. These 10 maps were manually selected to match the following cases: a combination with both thematic and geographic matches, only geographic matches, only thematic matches, and no match at all. Each participant was asked to compare the maps with their corresponding queries, and rank the degree of matching from 0 (not matching at all) to 5 (perfect matching). In total, we have collected 700 data records, and combined the results with the 900 matching feature scores (9 scores for each of the 100 maps). It is worth mentioning that a detailed study on user preferences is out of scope here and the topic has been extensively studied in the search and information retrieval literature. Here, we are only interested in estimating the relative importance of each matching feature to ensure cognitively meaningful results.

Based on the data from the human-participant experiment, we derive values for the coefficients in the regression model. To evaluate the necessity of including the thematic-geo interaction variable, we construct two regression models and will evaluate them respectively.

4.4 Evaluation

We evaluate the regression-based ranking model using the two criteria, namely the correctness of the ranking result and the convenience of being embedded into a SPARQL query.

To evaluate the correctness of ranking, we compare the ranking scores computed by the regression model with the average scores from human judgments. Two statistics, root-mean-square error (RMSE) and Pearson's correlation coefficient, have been used to quantify the closeness between the two. Figure 2 shows the comparison results.

The solid lines in the above two figures represent the reference line, and indicate the perfect consistence between the estimated ranking score and human judgments. The dotted lines represent the actual relation between the estimated and ground truth ranking. As can be seen, including the interaction variable brings higher correlation coefficient as well as lower RMSE. The dotted line in Fig. 2b is also closer to the reference line than that in Fig. 2a. The correlation coefficient in Fig. 2b is 0.7746 ($p < 0.01$) which indicates a high consistence between the estimated ranking and the average human judgments.

This regression-based ranking model can also be integrated into a Linked-Data-driven geoportal without much difficulty. To demonstrate this, we implement this

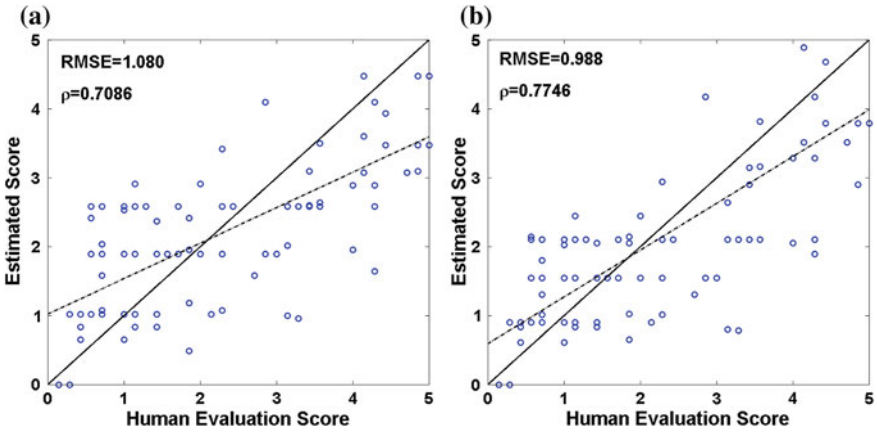


Fig. 2 Comparing estimated ranking scores with human judgments **a** without the interaction variable, **b** with the interaction variable

ranking model as a single SPARQL query (shown in Listing 1). Such a SPARQL query can be directly embedded into a system's existing search module without having to change other parts of the system.

```
SELECT ?item (COUNT(?titleThematicExact) AS ?TTE)
(COUNT(?titleThematicSimilar) AS ?TTS)
(COUNT(?descThematicExact) as ?DTE)
(COUNT(?descThematicSimilar) as ?DTS)
(COUNT(?titleGeoExact) as ?TGE)
(COUNT(?titleGeoSimilar) as ?TGS)
(COUNT(?descGeoExact) as ?DGE)
(COUNT(?descGeoSimilar) as ?DGS)
(((?TTE+?TTS+?DTE+?DTS)*(?TGE+?TGS+?DGE+?DGS)) as ?TGI)
(( λ1*?TTE + λ2*?TTS + λ3*?TGE + λ4*?TGS + λ5*?STE + λ6*?STS +
λ7*?SGE + λ8*?SGS + + λ9*?TGI) as ?ranking)
WHERE { OPTIONAL {
    ?item :hasTitleThematicTerm ?titleThematicExact .
    FILTER ( ?titleThematicKey = :exactThematicTerm ) }
  OPTIONAL {
    ?item :hasTitleThematicTerm ?titleThematicSimilar .
    FILTER ( ?titleThematicSimilar = :expandedThematicTerm ) }
  OPTIONAL {
    ?item :hasDescThematicTerm ?descThematicExact .
    FILTER ( ?descThematicExact = :exactThematicTerm ) }
  OPTIONAL {
    ?item :hasDescThematicTerm ?descThematicSimilar .
    FILTER ( ?descThematicSimilar = :expandedThematicTerm ) }
  OPTIONAL {
    ?item :hasTitleGeoTerm ?titleGeoExact .
    FILTER ( ?titleGeoExact = :exactGeoTerm ) }
```

```
        OPTIONAL {
         ?item :hasTitleGeoTerm ?titleGeoSimilar .
         FILTER ( ?titleGeoSimilar = :expandedGeoTerm ) }
        OPTIONAL {
         ?item :hasDescGeoTerm ?descGeoExact .
         FILTER ( ?descGeoExact = :exactGeoTerm ) }
        OPTIONAL {
         ?item :hasDescGeoTerm ?descGeoSimilar .
         FILTER ( ?descGeoSimilar = :expandedGeoTerm ) }
      } GROUP BY ?item ORDER BY Desc(?ranking) LIMIT 200
```

Listing 1: SPARQL query for estimating the relavance of resources and result ranking.

5 Flexible Queries for Knowledge Discovery

The existing Web API of ArcGIS Online only allows users to submit queries satisfying pre-designed templates. As a sample of ArcGIS Online metadata has been converted into Linked Data, they automatically support flexible queries without requiring additional hard coding in the Web API. In this section, we employ two scenarios to demonstrate these user-defined queries that can be performed, as well as some interesting knowledge that can be discovered. While only two queries are shown in this paper, we also provide more than 20 additional examples in the *knowledge discovery* menu of the implemented ArcGIS Online Linked Data Web interface (see Sect. 6).

5.1 Which Basemaps Are Popular?

ArcGIS Online allows users to browse the available basemaps, and records the number of times that each basemap has been **viewed** by users. Based on this number, one might assume that the most popular basemap is the one that has been viewed by most people. However, given the newly interlinked metadata, we can also count the times that each basemap has actually been **used**. In this scenario, we compare the results based on these two definitions of *popularity*. The below SPARQL query returns the result based on the times of views:

```
SELECT DISTINCT ?baseMap ?numViews
WHERE { ?baseMap arcgis:isBaseMapOf ?item .
        ?baseMap arcgis:numViews ?numViews }
ORDER BY DESC(?numViews) LIMIT 10
```

The result of the above query indicates that the *World Boundaries and Places* map has been **viewed** most frequently. However, making the number of usages as the criterion for popularity may lead to a different ranking. Below is the corresponding SPARQL query:

```
SELECT ?baseMap (count(distinct ?item) as ?usedTimes)
WHERE { ?baseMap arcgis:isBaseMapOf ?item }
GROUP BY ?baseMap
ORDER BY DESC(?usedTimes) LIMIT 10
```

Interestingly, the result indicates that the *World Topographic Map* is the one that have been **used** most times in other maps. In fact, it has been used 13,507 times which is significantly more than the usage of the *World Boundaries and Places* map (2855 times).

5.2 Which Group Has Created Most Maps About California?

In this scenario, we demonstrate the additional capabilities of GeoSPARQL, an OGC standard language for querying geographic RDF data. It allows users to query and summarize data based on not only non-spatial attributes but also geographic extents. As an example, we search for the user group that has created the highest number of maps about California.

```
SELECT DISTINCT ?group (count(?item) as ?itemCount)
WHERE { ?group arcgis:type arcgis:ArcGIS-Group .
        ?group arcgis:hasItem ?item .
        ?item  geo:hasGeometry ?itemGeo .
        ?itemGeo geo:asWKT ?wkt
        Filter (geof:sfWithin(?wkt, Polygon((-125 42, -120 42,
        -120 39, -114 34, -114 32,
        -120 32, -125 42))^^sf:wktLiteral))  }
GROUP BY ?group ORDER BY DESC(?itemCount) LIMIT 100
```

In the above query, we first define a polygon element to approximate the boundary of California. We then use this polygon as the extent limit for a geographic filter based on the topological relation *within*. ArcGIS Online items that fall within the boundary of California will be counted for each group, and the query returns the top 100 groups that have created maps about California. The result shows that the No.1 group is a Web GIS class from the University of California, Riverside.

6 Implementation

A prototypical Linked-Data-driven Web portal for ArcGIS Online has been implemented using the presented methods, and can be accessed via http://stko-exp.geog.ucsb.edu/linkedarcgis/. Based on the semantically annotated and enriched Linked Data, the portal provides the following capabilities:

- **Semantic Search**. This function enables the search of maps based on the enriched semantic interpretation of queries. Figure 3 shows an example of searching for *natural disasters in Utah*, in which the system returns maps about flood, tornado and other disasters. To increase the clarity of the search results, we use three columns to separately show maps that have both thematic and geographic matches, only thematic matches, and only geographic matches.
- **Knowledge Discovery**. This module shows additional examples for flexible queries automatically enabled by the Linked Data. We design a simplified user interface which deliberately hide the technique details, but interested users can click the *SPARQL* button to check the SPARQL statements used behind the scene.
- **GeoSPARQL**. This module demonstrates the capability of OGC's GeoSPARQL in supporting geographic queries on Linked Data. Users can search maps by inputting a thematic term (e.g., fire), and specifying a geographic area (e.g., California). While results will be shown as thumbnails, a map will also be shown at the bottom of the page to indicate the geographic extents of the search results.
- **Statistics**. This module gives a numeric summary of the exported and converted ArcGIS Online sample data.

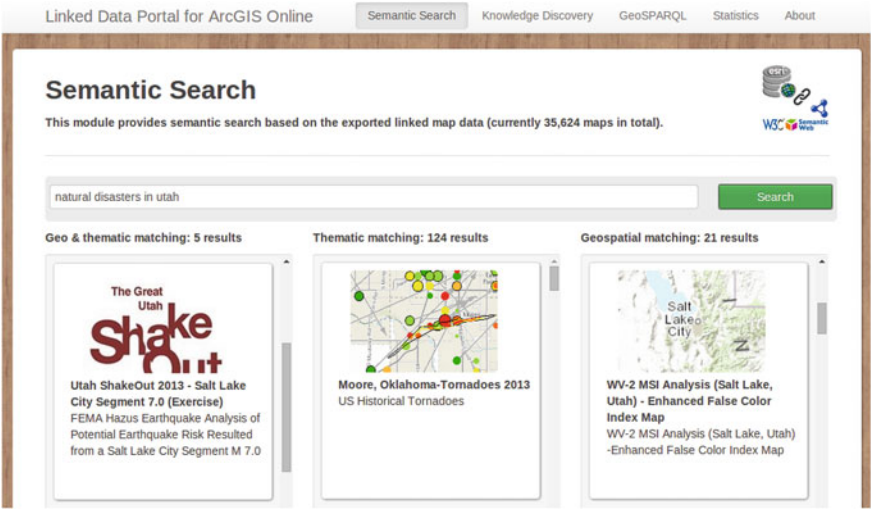


Fig. 3 A screenshot for the prototypical linked-data-driven web portal

7 Conclusions and Future Work

With the fast growth of online geoportals and the wide availability of GIS resources, there is an increasing demand for intelligent and flexible search to help users efficiently find data. Our work is an effort towards this direction. Based on ArcGIS Online, a large geoportal and online cloud service, we present a workflow for converting the metadata using the Linked Data principles and enriching them with machine-readable terms extracted from titles and descriptions. We design a semantic search function for Linked Data by expanding users' input queries and tuning a linear regression model with human participants. An evaluation experiment has been conducted to assess both the correctness and the usability of the presented semantic search function. As a sample of metadata from ArcGIS Online has been converted into Linked Data, they automatically support flexible queries without requiring pre-designing and hard-coding in a Web API. We use two scenarios to demonstrate the flexible queries that can be submitted to discover new knowledge from the data. An online prototype has been implemented using the presented methods as a proof-of-concept.

While we have taken a Linked-Data-driven approach in this work, it is worth to clarify that some techniques used in our workflow, such as query expansion and entity extraction, do not necessarily require a Linked Data approach. However, Linked Data automatically enables flexible and customized queries which significantly expand the searching capability that a GIS user can have. Thus, we consider Linked Data as an indispensable cornerstone in our solution. This research also has several limitations. For example, the coefficients of the regression model for semantic search are derived from a small number of participants. While the evaluation shows a fair performance in the search results, a larger scale human participant test is nevertheless necessary to further calibrate the model. In addition, since external services, such as *DBpedia Spotlight*, have been used to expand the queries in real time, the response time of the semantic search varies. While the online system is still a prototype, the search speed could be improved by integrating those external services as part of the entire system. Finally, so far we have extracted thematic concepts and geographic entities from the map titles and descriptions, and a next-step research could focus on inferring topic categories (e.g., whether this map is about *transportation* or *agriculture*) from the textual descriptions, thereby further enriching the voluntary metadata.

Acknowledgments This work is a collaborative effort from UCSB STKO Lab and ESRI Applications Prototype Lab. The authors would like to thank Jack Dangermond, Hugh Keegan, Dawn Wright, as well as the three anonymous reviewers for their constructive comments and feedbacks.

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In: *The semantic web* (pp. 722–735). Berlin, Springer.

Battle, R., & Kolas, D. (2012). Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web, 3*(4), 355–370.

Berners-Lee, T., Hendler, J., Lassila, O. (2001) *The semantic web* (pp. 29–37). Scientific American (2001).

Bizer, C., Heath, T., & Berners-Lee, T. (2009a). Linked data—the story so far. *International Journal on Semantic Web and Information Systems, 5*(3), 1–22.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., & Cyganiak, R. (2009b). DBpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(3), 154–165.

Butuc, M. G. (2009). Semantically enriching content using opencalais. *EDITIA, 9*, 77–88.

Dangermond, J. (2009). *GIS: Design and evolving technology*. ESRI, Fall: ArcNews.

Goodchild, M. F., & Glennon, J. A. (2010). Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth, 3*(3), 231–241.

Guha, R., McCool, R., Miller, E. (2003) Semantic search. In: *Proceedings of the 12th International Conference on World Wide Web*, (pp. 700–709). ACM.

Han, L., Kashyap, A., Finin, T., Mayfield, J., Weese, J. (2013) *UMBC ebiquity-core: Semantic textual similarity systems* (p. 44). Atlanta, Georgia, USA.

Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web: Theory and Technology, 1*(1), 1–136.

Hitzler, P., Krotzsch, M., Rudolph, S. (2011). *Foundations of semantic web technologies*. Boca Raton, CRC Press.

Hu, Y., Janowicz, K., McKenzie, G., Sengupta, K., Hitzler, P. (2013). A linked-data-driven and semantically-enabled journal portal for scientometrics. In: *The semantic web–ISWC* 2013 (pp. 114–129). Berlin, Springer.

Janowicz, K., Scheider, S., Pehle, T., & Hart, G. (2012). Geospatial semantics and linked spatiotemporal data–past, present, and future. *Semantic Web, 3*(4), 321–332.

Jones, C.B., Alani, H., Tudhope, D. (2001). Geographical information retrieval with ontologies of place. In: *Spatial information theory*, (pp. 322–335). Berlin, Springer.

Keßler, C., Janowicz, K., Kauppinen, T.: spatial@ linkedscience–Exploring the research field of GIScience with linked data. In: *Geographic Information Science* (pp. 102–115). Berlin, Springer (2012).

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*(2), 211.

Mendes, P.N., Jakob, M., Garca-Silva, A., Bizer, C. (2011). DBpedia spotlight: Shedding light on the web of documents. In: *Proceedings of the 7th International Conference on Semantic Systems* (pp. 1–8). ACM.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM, 38* (11), 39–41.

Tran, T., Cimiano, P., Rudolph, S., Studer, R. (2007). Ontology-based interpretation of keywords for semantic search. In: *The Semantic Web* (pp. 523–536). Berlin, Springer.

Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y. (2007) Spark: Adapting keyword query to semantic search. In: *The Semantic Web* (pp. 694–707). Berlin, Springer.

Real-Time Anomaly Detection from Environmental Data Streams

Sergio Trilles, Sven Schade, Óscar Belmonte and Joaquín Huerta

Abstract Modern sensor networks monitor a wide range of phenomena. They are applied in environmental monitoring, health care, optimization of industrial processes, social media, smart city solutions, and many other domains. All in all, they provide a continuously pulse of the almost infinite activities that are happening in the physical space—and in cyber space. The handling of the massive amounts of generated measurements poses a series of (Big Data) challenges. Our work addresses one of these challenges: the detection of anomalies in real-time. In this paper, we propose a generic solution to this problem, and introduce a system that is capable of detecting anomalies, generating notifications, and displaying the recent situation to the user. We apply CUSUM a statistical control algorithm and adopt it so that it can be used inside the Storm framework—a robust and scalable real-time processing framework. We present a proof of concept implementation from the area of environmental monitoring.

Keywords Big data and real-time analysis · Environmental sensor data · CUSUM · STORM

S. Trilles (✉) · Ó. Belmonte · J. Huerta
Institute of New Imaging Technologies, Universitat Jaume I, Castellón de la Plana, Spain
e-mail: strilles@uji.es

Ó. Belmonte
e-mail: belfern@uji.es

J. Huerta
e-mail: huerta@uji.es

S. Schade
Institute for Environment and Sustainability, European Commission—Joint Research Centre, Ispra, Italy
e-mail: sven.schade@jrc.ec.europa.eu

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_8

1 Introduction

Growing amounts of sensor networks measure almost every environmental and man-made phenomena we can think of. These networks can be of different types, but essentially they monitor particular physical characteristics. We can witness some of them in our daily lives, e.g. for environmental monitoring (meteorological, air quality, etc.), health care monitoring, industrial monitoring, or social monitoring/sensing. The Internet of Things (IoT) movement (Kortuem et al. 2010) has allowed these sensor networks to be connected to the Internet, and their access tends to get open to everybody, which ultimately allows to find and retrieve observations in large quantities—and this in every single second.

Each single sensor in each of these networks produces a stream of data and—depending on the particular refresh time—has the capacity to send a large number of measurements. As it becomes difficult to analyze all of these observations in the moment that the raw values are obtained (Manovich 2012), it is challenging to extract relevant knowledge in (near) real-time. This paper introduces this "Big Data" challenge and suggests a mechanism to analyze the arising flood of monitoring data.

Our work particularly addresses anomaly detection. As soon as anomaly is detected, we want to be able to launch a notification in order to (i) trigger a decision-making process, and (ii) inform about the anomaly that caused the event, together with surrounding context information. In many cases, this support has to be provided in real-time because decision-making is time-critical.

We present a system to analyze different sensor networks, to detect anomalies, and to send notifications. We base this system on the Storm framework,[1] which distributively and reliably processes unbounded streams of data in real-time. With this entirely new application of the Storm framework to the IoT, we become able to analyze multiple streams and apply dedicated anomaly-detection algorithms to it. In this pioneering work, we implement the CUmulative SUM (CUSUM) (Page 1954) algorithm in Storm. The resulting system can be applied to any series of values and determine anomalies in real-time. In its first implementation our system operates on environmental sensor networks, where we obtain a series of numerical values. The remainder of the paper is organized as follows. Section 2 shows the Storm framework and enumerates some of its usages. Section 3 presents the CUSUM algorithm and related work. Section 4 introduces the system's design. Section 5 presents a proof of concept for this work. The paper concludes in Sect. 6, which also includes pointers to future work.

[1]Storm, Apache Incubator, http://storm.apache.org.

2 Storm Framework

Storm is the central component of our proposed solution for processing and detecting anomalies from (big) data streams in real-time. It could be used to execute any algorithms on top of sensor data series. Section 2.1 provides a basic introduction to Storm, while Sect. 2.2 gives the technical background.

2.1 Introducing Storm

Storm was created at Backtype, a company that was acquired by Twitter in 2011, became an Apache Incubator project in September 2013, and finally reached the status of an Apache Top-Level Project in September 2014. It is a distributed real-time stream processing framework, which can be used to analyze, filter and normalize data series and apply many others regular-expression filters on data in real-time. The framework is fault tolerant and guarantees data processing.

Storm is comparable to Apache Hadoop (Shvachko et al. 2010). While Hadoop provides programmers a framework for performing batch operations, Storm provides a system for performing streaming computations, similar to Apache S4,[2] Samza,[3] Spark Streaming,[4] and others.

The Storm framework has already been used for a wide range of purposes:

- Simoncelli et al. (2013) used Storm to extract trending hashtags from Twitter, and count occurrences of citations.
- Sunderrajan et al. (2014) used it to detect outliers in the power consumption from smart plugs.
- Yutan et al. (2014) used Storm to monitor the flow of data and locate the source of attacks or anomalies in real-time by finding the specific IP addresses.
- Storm has also been used to apply data mining in real-time sources (De Francisci Morales 2013).
- Sitaram et al. (2013) applied Storm for speech recognition in real-time.

We suggest to use Storm in the context of the IoT, and especially for environmental monitoring, because it allows the finest-grained control of processing unit, offers greatest control over data streams, is mature, and has a strong community.

[2]Apache S4, Apache Incubator, http://incubator.apache.org/s4.

[3]Samza, Apache Incubator, http://samza.incubator.apache.org.

[4]Spark Streaming, Apache Incubator, http://spark.apache.org/streaming.

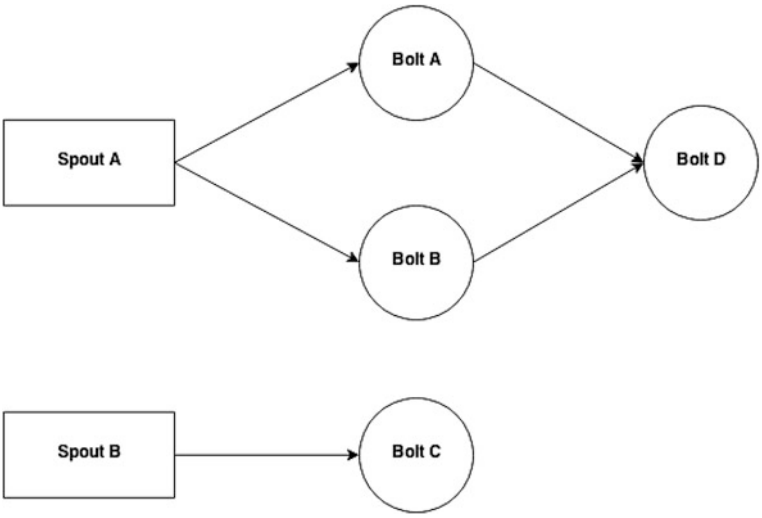


Fig. 1 An example of Storm *Topology* with different *Spouts* and *Bolts*

2.2 Technical Background of Storm

Storm has its own terminology, starting with the concept of *Topologies*. A *Topology* defines the way (workflow) in which a problem will be processed. It is similar to a job in Hadoop. However, Hadoop jobs will have an end, while the *Topology* will always run because there is a need for continuous computation.

Figure 1 illustrates a *Topology* as the workflow of *Spouts* and *Bolts*. The *Spouts* handle the insertion of data *tuples* into the *Topology* and send each of these *tuples* to *Bolts*. They thus connect the input stream and send the values to the *Bolt* that is responsible for the stream processing. Each *Bolt* processes the streams that it receives from the *Spout*. It applies any particular procedure to generate its output stream. The actual improvement of Storm compared to other solutions, is that these operations can be parallelized. Parallelisation is handled on the level of a single *Bolt* and the parallelization factor is defined as part of the *Topology*.

Storm offers an at-least-once processing guarantee, but does not consider the order in which data streams are emitted. In fact, the *tuples* will have a different order when they are processed. For the objective of this work is necessary to maintain the order of the *tuples*. We use Trident[5] to ensure exactly this. Trident is a high-level abstraction framework for computation on Storm. As with the core Storm Application Programming Interface (API), Trident uses *Spouts* as the source of data streams. It has consistent, exactly-once semantics (same order), so it is easy to

[5]Storm Trident, Apache Incubator, http://storm.apache.org/documentation/Trident-API-Overview.html.

reason about Trident *Topologies*. Trident already offers different operations, such as functions, filters and create aggregations from streams of *tuples*.

3 CUmulative SUM (CUSUM)

We selected the CUSUM algorithm for detecting anomalies in data series from environmental monitoring. In essence, this algorithm compares two different instances of the same discrete probability function for a data series (Mesnil and Petitgas 2009). The algorithm was initially developed for industrial control purposes. In recent years, it has been successfully used in other areas. An overview of past applications is provided in Sect. 3.1. The details of the CUSUM algorithm are depicted in Sect. 3.2.

3.1 Applications of CUSUM

Osanaiye and Talabi (1989), for instance, used the CUSUM algorithm in order to detect possible outbreaks of epidemics. Grigg et al. (2003) analyzed the 30-day mortality for patients after cardiac surgery. Furthermore, CUSUM used is to improve the communication in Wireless Sensor Networks. Jeske et al. (2009) developed two CUSUM change-point detection algorithms for data network monitoring applications. CUSUM has additionally been used in pattern recognition, specifically in neural networks. Sample of them is Guh and Hsieh (1999) study where it proposes an artificial neural network based model, which contains several back propagation networks. Chuen-Sheng (1995) described an alternative approach for statistical process control, using artificial neural network technology and compares its performance with that of the combined Shewhart-CUSUM schemes.

More recently, CUSUM has been adapted and used for a number of environmental problems, including the following works:

- Barratt et al. (2007) used the CUSUM algorithm to detect anomalies in carbon monoxide (CO) levels after the introduction of a permanent bus line in Marylebone Road, London.
- Carslaw et al. (2006) presented an analysis of the same data source as the Barratt et al. (2007) study. They analyze various components with CUSUM, such as nitrogen oxides (NOx), nitrogen dioxide (NO_2), particulate matter with diameter less than 10 μm (PM10), particulate matter with diameter less than 25 μm PM25, and PMcoarse (defined as PM25-10).
- Chelani (2011) compares a modified CUSUM algorithm with the original to detect anomalies in phenomena, such as NO_2, CO and PM10. The paper concludes that the modified CUSUM can help in detecting anomalies when there is greater variability in the observations.

- Charles and Jeh-Nan (2002) proposed to use the CUSUM control chart to monitor data about industry emissions to the environment to detect abnormal changes in a timely manner.

Following these success stories, we decided to use CUSUM as part of our anomaly detection system. In our proof of concept implementation (Sect. 5) CUSUM is applied to anomaly detection of air pollutants.

3.2 Technical Background of CUSUM

CUSUM considers a set of observations (x_i) with collected observation $i = 1, \ldots, n$, where n is the number of data points. The algorithm assumes that these observations are in-control when the collection has a mean (μ) and standard deviation (σ^2) for a normal period and following a normal distribution $N(\mu, \sigma^2)$. When the process is in-control, we can obtain the CUmulative SUM (S_i) in an iterative way through the following expression:

$$S_i = S_{i-1} z_i \tag{1}$$

where $S_0 = 0$, z_i is the standard normal variable, $z_i = \frac{x_i - \bar{x}}{s}$, $\bar{x}$ is the mean and s is the standard deviation of the time series. Further, the change in terms of increased or decreased process mean can be detected, respectively by computing the quantities as (Lucas 1982):

$$\begin{aligned} S_{H_i} &= MAX[0, (z_i - k) + S_{H_i - 1}] \\ S_{L_i} &= MIN[0, (z_i + k) + S_{L_i - 1}] \end{aligned} \tag{2}$$

where the parameter k is the reference value to be chosen appropriately. The parameter, k, is the allowable "slack" in the process and is usually set to be one half of the mean one wishes to detect. The confidence limits (*threshold*) specified for the CUSUM control charts are $\hat{\mathrm{A}} \pm h\sigma_x$, where $h = 5$ and σ_x is the standard deviation (Barratt et al. 2007).

When S_{H_i} or S_{L_i} overcome the *threshold*, the algorithm detects an anomalies. If S_{H_i} exceeds the *threshold* the anomaly will be due to the increase (*up-event*). And If S_{L_i} is greater than the *threshold*, it will be due to the decrease (*down-event*).

Two characteristics of CUSUM limit the sensitivity of results. First, the identification of an out-of-control process relies on the assumption that readings are statistically independent and follow a normal distribution. Second, phenomenon measurements can have some seasonality and long-term trends. This has the effect that the *threshold* may be out of adjustment.

4 System Design

This section presents our system design for real-time anomaly detection. This system is formed by four components: data streams from sensor networks; the Real-time Message Service (RMS) (that equally handles the input as well as the output data flows); *Topology* developed in Storm, including our implementation of the CUSUM algorithm for real-time anomaly detection; and the application to visualize the final outcomes. Figure 2 shows these components and the connections between them. The follow subsections detail the central components.

4.1 Real-Time Message Service

We have to handle observations in real time, i.e. as soon as new values become available on the Web. Traditionally, web-based resources are accessed by client's HTTP requests to a server, which then responds by returning the resource. This procedure should be repeated every time that the user wants to access the resource. For data sources from high refresh rates, such as stock price or environmental sensor data, require frequent requests to the server in order to (almost) constantly receive the latest data sets.

More effective and efficient approaches have been developed to address such cases. They are based on polling mechanisms, where the client repeatedly sends new requests to the server. If the server has no new data, then it sends appropriate indication and closes the connection. The client then waits a bit and sends another request after some time.

A more recent approach is long-polling. In this case, the client sends a request to the server and the server keeps the connection open for a previously defined period, or a timeout period. When the server has new data available, then it directly transmits it to the client. An example of this approach is the Message Oriented

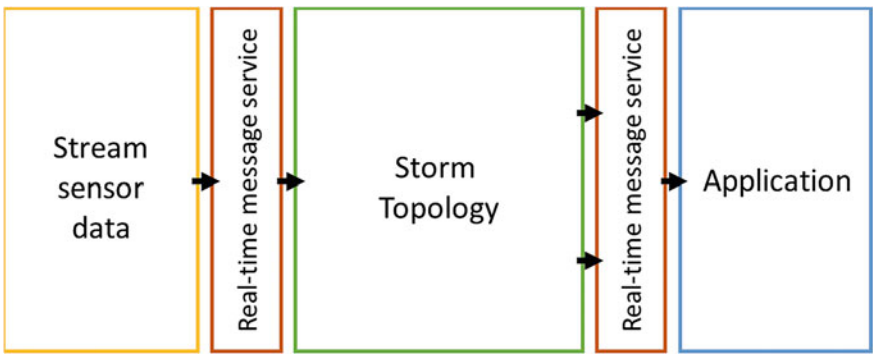


Fig. 2 System design for the anomalies detection

Middleware, which provides an infrastructure to send and receive messages between distributed systems.

We decided to use this approach in order to realize the real-time communication in our system. This component is called RMS. It is used to serve the data provided by the sensors and is based in Java Message Service (JMS) (Hapner et al. 2002) to deliver asynchronous communication via the web. JMS is a standard for the implementation of Message Oriented Middleware on the Java platform for sending messages between two or more parts. In the JMS context, exists different components:

- *Provider*: an implementation of the JMS interface for a Message Oriented Middleware.
- *Client*: an application or process that produces and/or receives *Messages*.
- *Producer/Publisher*: a *Client* that creates and sends *Messages*.
- *Consumer/Subscriber*: a *Client* that receives *Messages*.
- *Message*: a *Message* can be any object or data that needs to be transported using JMS.
- *Queue*: a staging area that contains *Messages* that have been sent and are waiting to be read (by only one *Consumer*). A *Queue* only guarantees that each *Message* is processed only once.
- *Topic*: a distribution mechanism for publishing *Messages* that are delivered to multiple *Subscribers*.

JMS has two different communication models, which are both relevant for our work (see also Fig. 3):

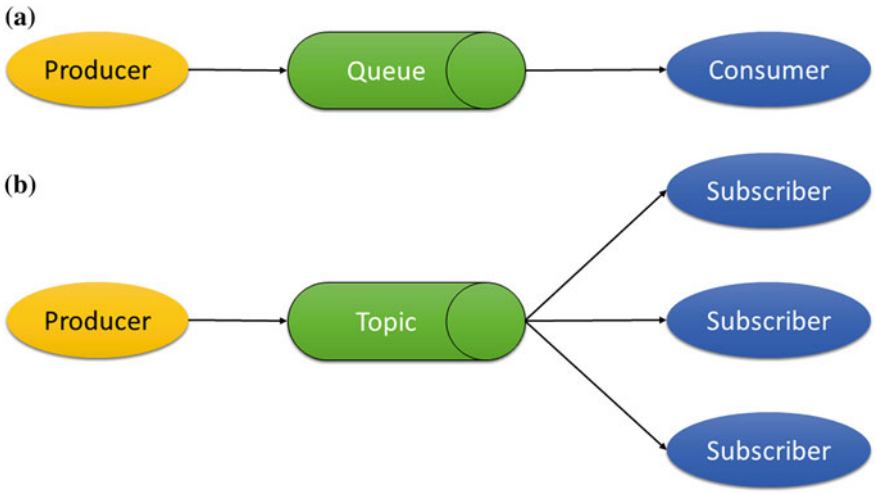


Fig. 3 JMS communication models. **a** *Point-to-point* and **b** *Publish/Subscribe*

- *Point-to-point* model: in this model, the *Messages* are routed to an individual *Consumer* which maintains a *Queue* of "incoming" *Messages* (Fig. 3a).
- *Publish/subscribe* model: this model supports publishing *Messages* to a particular *Message Topic*. *Subscribers* may register interest in receiving *Messages* on a particular *Message Topic* (Fig. 3b).

For the development of the RMS, we have used a JMS framework called ActiveMQ.[6] ActiveMQ is a popular and powerful open source persistence messaging and integration patterns server with scheduler capabilities, acts as a message broker in the framework. It support different protocols, such as: OpenWire,[7] REST (Fielding 2000), Simple (or Streaming) Text-Oriented Messaging Protocol (STOMP),[8] Web Socket (Hickson 2011) and more. For our work we used a STOMP interface. It is a simple text-oriented protocol, similar to HTTP. STOMP provides an interoperable wire format that allows clients to communicate with almost every available message broker.

We provide a *Client*, called JMS *Client*, to connect to RMS via the STOMP interface. It offers the different models of connections (both *Point-to-point* model as *Publish/subscribe*). This *Client* can be to perform as *Producer* or *Consumer*. It will be used into the stream sensor data, the Storm *Topology* and the application.

4.2 Stream Sensor Data

We build on our previous activities (Trilles et al. 2014) in order to access sensor data sources. In that work, wrapping techniques were applied to obtain sensor observations from each individual sensor. For the work at hand, we re-use these techniques, but now serve the observations via real-time interfaces inside the RMS. Each time when a new observation is produced by a sensor, it becomes directly distributed by the RMS via its real-time interface. At this occasion, we apply the *Point-to-point* model of JMS, because it ensures that the receiver will get all the produced *Messages*.

In order to connect with the RMS, the JMS *Client* (introduced in the Sect. 4.1) has been used as a *Producer*. Any new observation is sent to the corresponding *Queue*. One *Queue* per phenomenon and sensor. The Fig. 4 illustrates this use of the RMS.

To indicate the correct parameters to connect to each *Queue*, an eXtensible Markup Language (XML) file has been created. It contains a single entry to define the sensor network, contains details about each sensor: an identifier, name, and city, state or location. The sensor entry includes a separate element for each phenomenon that is measured by the sensor. Each of these elements contains details about the

[6]ActiveMQ framework, Apache Software Foundation, http://activemq.apache.org.

[7]OpenWire protocol, Apache Incubator, http://activemq.apache.org/openwire.html.

[8]STOMP protocol, Apache Incubator, http://activemq.apache.org/stomp.html.

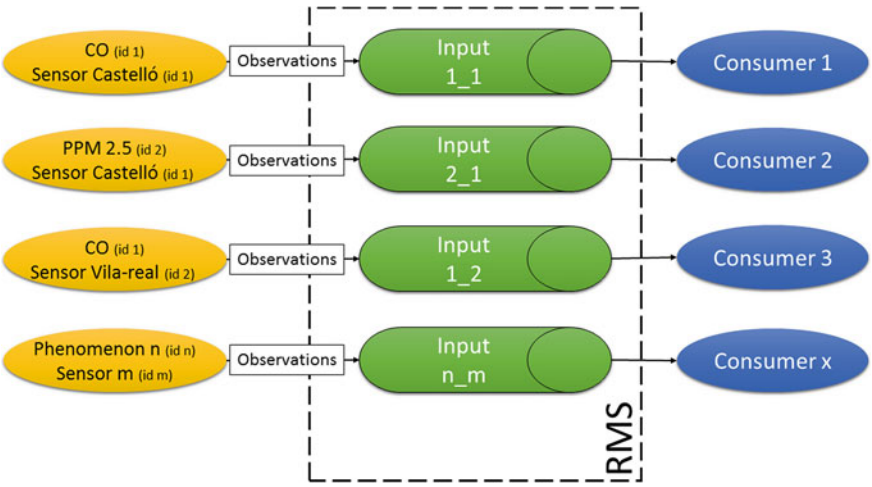


Fig. 4 The figure shows how RMS is used for the stream sensor data

measured phenomenon: an identifier, observed property, unit of measure. It also contains the parameters that are needed to run CUSUM (threshold and k).

A proprietary format has been used for encoding each observation. This observation contains an identifier, the measured value, the time-stamp and the sensor identifier. It is encoded using JavaScript Object Notation (JSON).

4.3 Storm Topology

Now that the real-time interfaces are available to serve a sensor data, we can concentrate on the *Topology* for anomaly detection. Within the *Topology* definition, we call the sensor observations *tuples*. The *Bolts* are responsible for processing these *tuples*. In our case, the *Topology* consists of one *Spout* and two separated *Bolts* (Fig. 5). We visit each of them in detail below.

The *Spout* is called *Real-time Spout* and aims to connect with the RMS to obtain the *tuples*. In this *Spout* the JMS *Client* has been used as *Consumer*. As soon as the RMS provides a new observation of the stream, it is read by the *Real-time Spout*. In the *Point-to-point* model, when the *Spout* reads a *tuple*, it is deleted from the *Queue*. This *Spout* is responsible for transforming the *Message* and creating the *tuples* that will be passed to the *Bolts*. It also uses the information from the XML files to connect to the RMS.

The first *Bolt*, called *Observations Bolt*, is only responsible for providing the latest *tuples* that the *Spout* has sent. The *Observation Bolt* is necessary to serve the *tuples* in a uniform way and these can be consumed by the final applications. It has two different functionalities. First, when the *Bolt* receives a new *tuple* from the

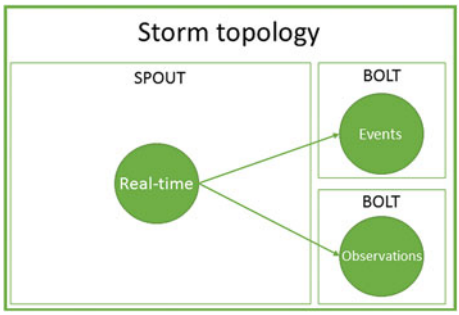


Fig. 5 The *Topology* created to apply CUSUM in the sensor data

Real-time Spout, it sends the observation using the RMS. Inside the *Bolt*, the JMS *Client* has been used as *Producer* to connect with the RMS. The *Observations Bolt* creates different *Queues* per phenomenon and sensor. Secondly, this *Bolt* serves a set of the latest *tuples* that were sent by the *Real-time Spout*. A First-In, First-Out (FIFO) buffer is used to store a few previous *tuples*—one for each phenomenon that the system supports. When receives a new *tuples*, the *Observations Bolt* adds the *tuple* to the buffer and the oldest *tuple* is removed. In this way, the user can access a small range of previous observations and does not see only the last observation. These observations are also offered using the RMS. The functionalities are both realised with the *Publish/Subscribe* model, so that different *Consumers* (final applications) can connect to the same *Queue*.

The *Events Bolt* is responsible to apply CUSUM to the series of *tuples* that are provided by the *Spout*. As already mentioned, Storm only ensures that all *tuples* are processed, but it does not guarantee the processing order of the *tuples* will be the same with that have read. Trident has been used to solve this inconvenience.

In order to execute the CUSUM algorithm on each phenomenon-specific measurement stream (as described in Sect. 3.2), we separate the tuples, which arrive from the sensor network, using unique phenomenon identifiers. The pseudocode below resumes the real-time variation for CUSUM. For the execution, we facilitate a data structure that cumulates and stores the sum for each iteration. Finally, we evaluate if the recent *tuple* causes an event by using a dedicated threshold.

When an anomaly is detected by CUSUM, the *Events Bolt* uses the JMS *Client* to send notifications to the RMS. Again, a *Queue* is created for each phenomenon and sensor. We again apply the *Publish/Subscribe* model (Fig. 6).

The events are currently also provided in a proprietary format with JSON encoding. Each event contains a sensor identifier and the identifier of the particular observation that has caused the event, together with a string value that indicates the identified event refers to an exceedance of the threshold (up-event) or to the fact that the observed value falls below the thresholds (down-event).

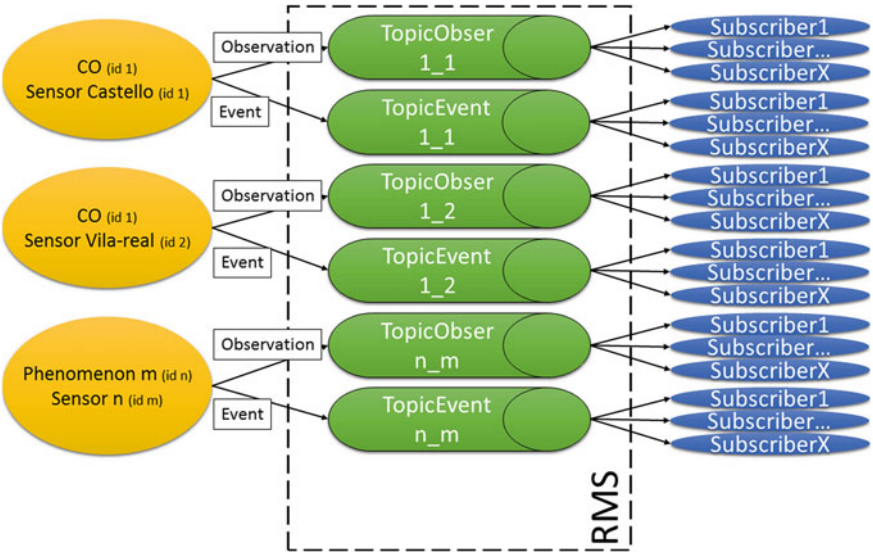


Fig. 6 The figure shows how is used RMS for the outputs, both for observations and events

4.4 Application Able to Connect to the RMS

An application has been designed in order to visualize the RMS outputs. The application has a single main use case (*Sensor map*) to visualize the real-time observations and the events (Fig. 7). This use case shows the sensors provided from the XML file through the *Sensor data sensor settings*.

To obtain the RMS outputs, the *Sensor map* exclude two sub-use cases, *Sensor observation* and *Sensor event*. The former is responsible to connect to the RMS and

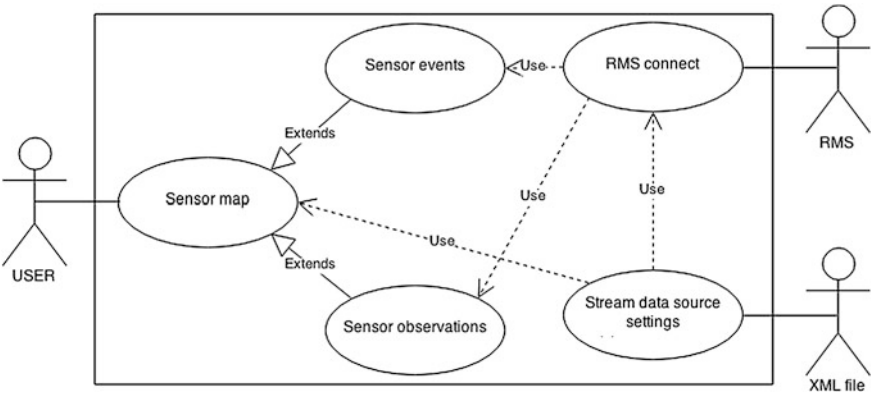


Fig. 7 The use case diagram for the designed application

retrieve the historical and last observations. The latter connects to the RMS for obtaining. The *Sensor event* connects to the RMS for obtaining the generated events. Both use cases apply the *RMS connect* as a *Subscriber* JMS *Client* in order to establish the connection with the RMS and obtain the *Topics* from the different subscribed *Queues*. This use case obtains the settings, which are required to connect with the RMS, from the *Sensor data sensor settings*.

Algorithm 1 CUSUM algorithm for real-time

procedure CUSUM REAL-TIME
  $numPhen =$ num of phenomena in the system
  $S_{previous_{high}}[numPhen] = 0$
  $S_{previous_{low}}[numPhen] = 0$
  while new observation $! = false$ **do**
    $observation =$ value of the new observation
    $idPhen =$ identifier of the phenomenoun
    $SNV = \frac{observation - \mu[idPhen]}{\sigma[idPhen]}$
    $S_{current_{high}}[idPhen] = MAX[0, (SNV - k[idPhen]) + S_{previous_{high}}[idPhen]]$
    $S_{current_{low}}[idPhen] = MIN[0, (SNV + k[idPhen]) + S_{previous_{low}}[idPhen]]$
    if $S_{current_{high}}[idPhen] >= threshold[idPhen]$ **then**
      Send "Up-Event"
      $S_{current_{high}}[idPhen] = 0$
      $S_{current_{low}}[idPhen] = 0$
    end if
    if $S_{current_{low}}[idPhen] >= threshold[idPhen]$ **then**
      Send "Down-Event"
      $S_{current_{low}}[idPhen] = 0$
      $S_{current_{high}}[idPhen] = 0$
    end if
    $S_{previous_{high}}[idPhen] = S_{current_{high}}[idPhen]$
    $S_{previous_{low}}[idPhen] = S_{current_{low}}[idPhen]$
  end while
end procedure

5 Proof of Concept

We implemented a proof of concept for the proposed system. The air quality network of the Valencian Community government[9] is used for testing purposes and an event dashboard illustrates the outcomes. Figure 8 shows all involved components.

[9] http://www.citma.gva.es/web/calidad-ambiental/datos-on-line.

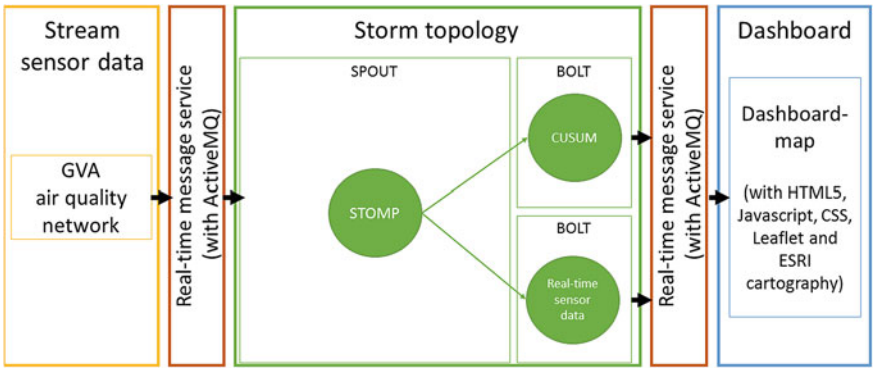


Fig. 8 The system design details for the proof of concept

5.1 Example Dataset

The Valencia Community government has deployed a network of 61 active stations for measuring the air quality in the region (Fig. 9). Its stations measure levels of multiple pollutants and meteorological conditions in urban, rural and industrial areas. These pollutants include sulphur dioxide (SO_2), nitrogen monoxide (NO), nitrogen dioxide (NO_2), nitrogen oxides (NOx), carbon monoxide (CO), ozone (O_3), benzene (C_6H_6) and other hydrocarbons such as toluene and xylene; particulate matter with diameter less than 10 μm (PM10), 2.5 μm (PM2.5), and 1 μm (PM1). The sensor network also covers metal levels such as arsenic, nickel, cadmium, lead and polycyclic aromatic hydrocarbons on the PM10 fraction. Some stations are also equipped with meteorological sensors for measuring parameters, such as wind speed and direction, relative humidity, solar radiation, atmospheric pressure and precipitation. The measurements are published on a website that is updated hourly. In this way, this data source provides both: historical and real-time data, which we need to test the proposed method.

As earlier commented, to use the CUSUM algorithm we need two parameters (threshold and k) per phenomenon that our system analyses. We obtain these parameters with historical data from the presented data source. For this, we have used one year of historical data (1st January 2013 to 31st December 2013). To apply CUSUM, and following the work of others (see Sect. 3.1), we are considering that all analyzed phenomena follow a normal distribution.

5.2 Event Dashboard

We developed an event dashboard to present the data provided by the RMS. All sensors of a network can be displayed on a map using markers (Fig. 10a). Inside the

Fig. 9 Map of the 61 air quality stations in the Valencia network

marker appears the amount of events that have been caused by this particular sensor. If this sensor triggers an event it appears in red. The dashboard does not differ whether the event is "up" (exceedance of the threshold) or "down" (falling under the threshold). A scale clustering has been applied to the markers following the quantity of events (Fig. 10b). When zooming out, the markers will be combined to the total amount of events that have been launched inside the cluster. The colour of the marker will be red if one sensor of this cluster launched an event.

When a user selects a single sensor marker, new markers appear as a menu (Fig. 11a). Each new marker symbolizes a phenomenon that is associated to this particular sensor, the names of the phenomena and the marker will appear in red, if a new event has been reported. Upon mouse click on one of the phenomena marker, the dashboard displays a pop-up widget that displays the latest observations in a graph (Fig. 11b). These observations are obtained from the buffer of the RMS. The graph is dynamically updated with the latest observations from the RMS. In the graph, events are highlighted as red rhombus. Also, these events are obtained from the RMS. The chart can display each of the observations interactively. The user can display different graphs simultaneously, even different phenomena from different sensors. In this way, one can compare the values of for same phenomena inside the same network.

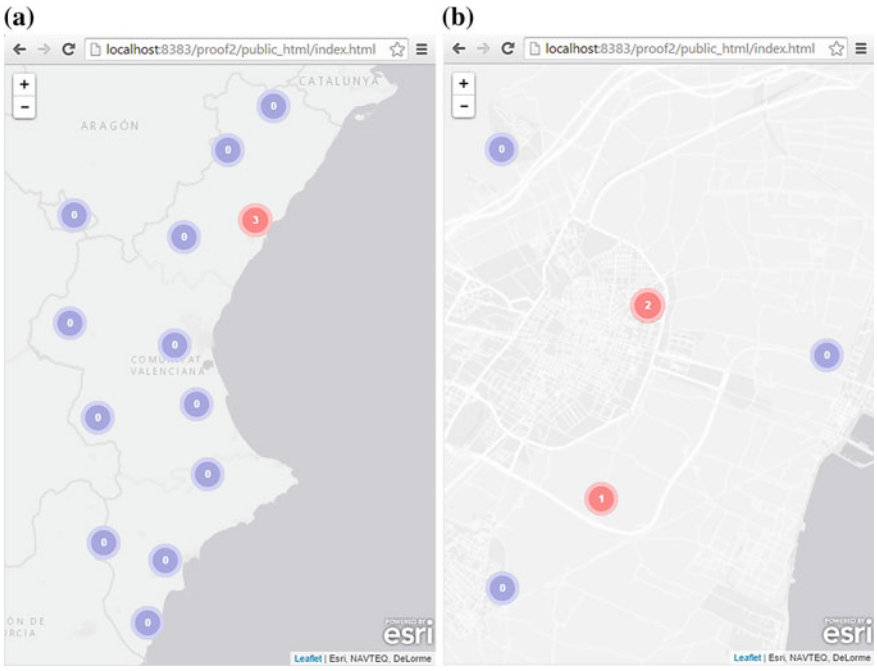


Fig. 10 **a** It shows the sensors (or cluster) as markers and the events are indicated inside the marker. **b** The figure compared with (**a**), shown as applied clustering by the amount of launched events

In order to implement a flexible, compatible and standards-compliant solution, we re-used a combination of already existing frameworks:

- Leaflet[10] with ESRI cartography,[11] to put the makers on the map. It proved to be fast and efficient. In addition, it can execute in a restrictive environments, such as smartphones.
- Another library that we used is Bootstrap.[12] It offer the capacity to building a responsive dashboard, can adapt to the device features. Also, we use jQuery [13] to handle popups.
- Finally other framework used is Highchart JS.[14] It is a graphics library written in HTML5 and JavaScript. The library provides an easy and interactive way to generate graphs in a web environment.

[10]Leaflet: An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps, http://leafletjs.com.

[11]ESRI, http://www.esri.com.

[12]Bootstrap, Twitter http://getbootstrap.com.

[13]jQuery, jQuery Foundation http://jquery.com.

[14]Highcharts JS, Highcharts AS http://www.highcharts.com.

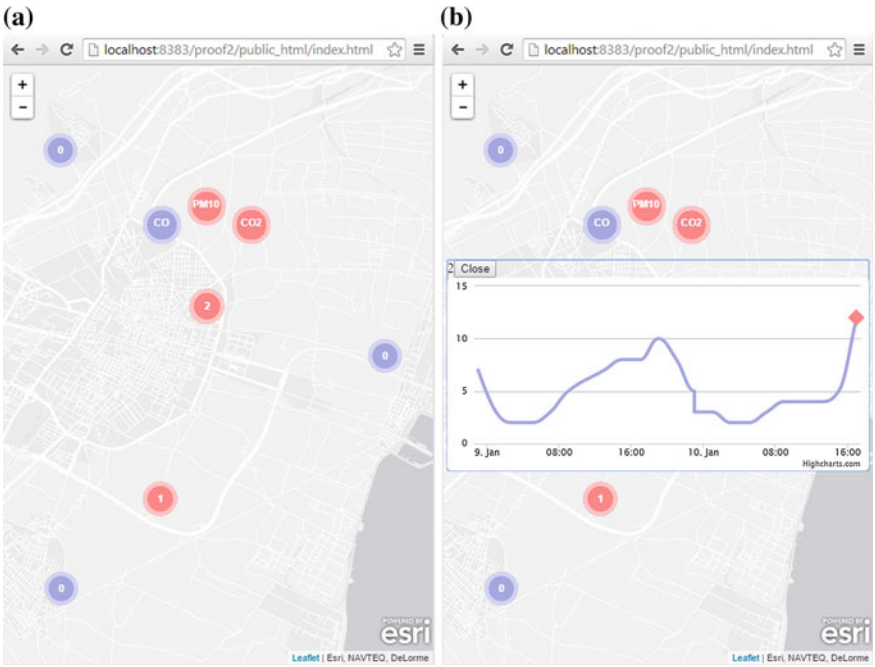


Fig. 11 **a** It shows A menu to visualize phenomena is show in each marker. **b** A pop-up with a graph are displayed to visualize the last observations and the events launched

6 Conclusions

The ever-increasing amount of sensors challenges the real-time detection of anomalies (Morris and Paradiso 2002). We propose a methodology and a system to analyze data streams from sensor networks near to real-time. This paper presented our system design in which we use novel technologies, such as the stream processing framework Storm, and Java Message Service (JMS). To our knowledge this is the first time that Storm is applied in the IoT and environmental monitoring.

In order to detect events in the streams of observations, we have developed a variation of CUSUM algorithm (Page 1954) so that it can run inside the Storm framework. Apart from the overall anomaly-detection solution, we also developed a dashboard application to visualize events that have been detected by the system, together with the contextual sensor data. A proof of concept implementation illustrated the overall feasibility of the approach using an example data source from environmental monitoring. CUSUM has strict underlying assumptions on value distributions, but it provides good results when it is applied on this type of data (Barratt et al. 2007; Barratt and Fuller 2014).

The primary challenge of the presented work was to provide a system able to analyze data from sensors and detect abrupt changes in a near the real-time.

Our system allows to analyze each of the observations without faults. Although, the used data source has low refresh rates, it could scale to higher refresh rates or add multiple data sources into the same system. As Storm was already successfully used in various other application areas, we are confident that integrations can be realised on top of this framework.

The second challenge was to detect abrupt changes in observation series. CUSUM has proven to be useful in looking at anomalies in the series of observations of air quality and weather. Although, we have validated our proposal with a particular data source, our overall system is designed to work simultaneously with multiple data sources of diverse characteristics, either with different kinds of interface connection or different data encodings. However, CUSUM presents some limitations that must be taken into account, such as the consideration of all the series follow a normal distribution and a series of observations cannot have trend changes. The use of alternatives algorithms, for example the one developed by Chelani (2011), remains to be investigated. This should particularly consider those algorithms that can account for phenomenon-specific probability distributions.

Now that the overall work-flow has been put into place, we concentrate on the standards support of the involved components. In the next step, we migrate the exchange formats to the Observations and Measurements standard (O&M) (Cox 2007) of the Open Geospatial Consortium (OGC). This is most important for encoding the output of the event detection components in a way that it can be easily integrated into diverse application tools, and to ensure interoperability with third-party systems.

Considering the input that is coming from any kind of sensor that produces values in metric scales, we will also soon support O&M as a format. In the medium term, we plan to extend the input related part of the system beyond O&M. We plan to apply a brokering solution (Buschmann et al. 1996; Nativi et al. 2012), so that inputs of multiple types can be accepted. In this way, the overall system will be able to operate on the most common standards for encoding measurements, but remains flexible for adoption new (even proprietary) formats.

We also plan to test the scalability of the system. This will be using a larger number of sources and a higher refresh rates. These experiments should have not affect on performance because Storm has been explicitly designed for scalability in order to be able to deal with huge amounts of data.

Acknowledgments This work has been supported in part by European Commission and Generalitat Valenciana government (grants ACIF/2012/112 and BEFPI/2014/067).

References

Barratt, B., & Fuller, G. (2014). Intervention assessments in the control of PM10 emissions from an urban waste transfer station. *Environmental Science: Processes and Impacts, 16*(6), 1328–1337.

Barratt, B., Atkinson, R., Anderson, H., Beevers, S., Kelly, F., Mudway, I., & Wilkinson, P. (2007). Investigation into the use of the cusum technique in identifying changes in mean air pollution levels following introduction of a traffic management scheme. *Atmospheric Environment, 41*(8), 1784–1791.

Buschmann, F., Meunier, R., Rohnert, H., & Sommerlad, P. (1996). *Pattern-oriented software architecture: A system of patterns* (Vol. 1). New York: Wiley.

Carslaw, D., Ropkins, K., & Bell, M. (2006). Change-point detection of gaseous and particulate traffic-related pollutants at a roadside location. *Environmental Science and Technology, 40*(22), 6912–6918.

Charles, J., & Jeh-Nan, P. (2002). Evaluating environmental performance using statistical process control techniques. *European Journal of Operational Research, 139*(1), 68–83.

Chelani, A. (2011). Change detection using cusum and modified cusum method in air pollutant concentrations at traffic site in Delhi. *Stochastic Environmental Research and Risk Assessment, 25*(6), 827–834.

Chuen-Sheng, C. (1995). A multi-layer neural network model for detecting changes in the process mean. *Computers and Industrial Engineering, 28*(1), 51–61.

Cox, S. (2007). Observations and measurements part 1—observation schema. Technical report, Open Geospatial Consortium (OGC).

De Francisci Morales, G. (2013). Samoa: A platform for mining big data streams. In *Proceedings of the 22nd International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, WWW '13 Companion* (pp. 777–778).

Fielding, R. (2000). Representational state transfer (rest) (Chap. 5). Fielding Dissertation.

Grigg, O., Farewell, V., & Spiegelhalter, D. (2003). Use of risk-adjusted CUSUM and RSPRTcharts for monitoring in medical contexts. *Statistical Methods in Medical Research, 12*(2), 147–170.

Guh, R., & Hsieh, Y. (1999). A neural network based model for abnormal pattern recognition of control charts. *Computers and Industrial Engineering, 36*(1), 97–108.

Hapner, M., Burridge, R., Sharma, R., Fialli, J., & Stout, K. (2002). *Java message service*. Santa Clara, CA: Sun Microsystems Inc.

Hickson, I. (2011). The WebSocket API. W3C Working Draft WD-websockets-20110929, September.

Jeske, D., Montes De Oca, V., Bischoff, W., & Marvasti, M. (2009). Cusum techniques for timeslot sequences with applications to network surveillance. *Computational Statistics Data Analysis, 53*(12), 4332–4344.

Kortuem, G., Kawsar, F., Fitton, D., & Sundramoorthy, V. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing, 14*(1), 44–51.

Lucas, J. (1982). Combined Shewhart-CUSUM quality control schemes. *Journal of Quality Technology, 14*(2).

Manovich, L. (2012). Trending: The promises and the challenges of big social data. In M. K. Gold (Ed.), *Debates in the digital humanities* (pp. 460–475). Minneapolis: U of Minnesota P.

Mesnil, B., & Petitgas, P. (2009). Detection of changes in time-series of indicators using cusum control charts. *Aquatic Living Resources, 22*(02), 187–192.

Morris, S., & Paradiso, J. (2002). Shoe-integrated sensor system for wireless gait analysis and real-time feedback. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint, IEEE* (Vol. 3, pp. 2468–2469).

Nativi, S., Craglia, M., & Pearlman, J. (2012). The brokering approach for multidisciplinary interoperability: A position paper. *International Journal of Spatial Data Infrastructures Research, 7*, 1–15.

Osanaiye, P., & Talabi, C. (1989). On some non-manufacturing applications of counted data cumulative sum (CUSUM) control chart schemes. *The Statistician, 38*(4), 251–257.

Page, E. (1954). Continuous inspection schemes. *Biometrika, 41*(1/2), 100–115.

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), IEEE* (pp. 1–10).

Simoncelli, D., Dusi, M., Gringoli, R., & Niccolini, S. (2013). Stream-monitoring with blockmon: Convergence of network measurements and data analytics platforms. *SIGCOMM Computer Communication Review, 43*(2), 29–36.

Sitaram, D., Srinivasaraghavan, H., Agarwal, K., Agrawal, A., Joshi, N., & Ray, D. (2013). Pipelining acoustic model training for speech recognition using storm. In *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation (CIMSim)* (pp. 219–224).

Sunderrajan, A., Aydt, H., & Knoll, A. (2014). Real time load prediction and outliers detection using storm. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, ACM, New York, NY, USA, DEBS '14* (pp. 294–297).

Trilles, S., Belmonte, O., Diaz, L., & Huerta, J. (2014). Mobile access to sensor networks by using GIS standards and restful services. *IEEE Sensors Journal, 14*(12), 4143–4153.

Yutan, D., Jun, L., Fang, L., & Luying, C. (2014). A real-time anomalies detection system based on streaming technology. In *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 275–279).

Towards Real-Time Processing of Massive Spatio-temporally Distributed Sensor Data: A Sequential Strategy Based on Kriging

Peter Lorkowski and Thomas Brinkhoff

Abstract Sensor data streams are the basis for monitoring systems which infer complex information like the excess of a pollution threshold for a region. Since sensor observations tend to be arbitrarily distributed in space and time, an appropriate interpolation method is necessary. Within geostatistics, kriging represents a powerful and established method, but is computation intensive for large datasets. We propose a method to exploit the advantages of kriging while limiting its computational complexity. Large datasets are divided into sub-models, computed separately and merged again in accordance with their kriging variances. We apply the approach to a synthetic model scenario in order to investigate its quality and performance.

Keywords Continuous phenomena · Sensor data streams · Spatio-temporal interpolation · Kriging · Deviation map

1 Introduction

The monitoring of continuous environmental phenomena such as temperature, air pollution or radiation is becoming an increasingly important issue in science. Low-cost wireless sensors and an efficient communication infrastructure become available more than ever. Both are the prerequisites for real-time environmental monitoring systems. However, appropriate strategies to deal with massive sensor data describing continuous spatio-temporal fields are still evolving (Appice 2014; Whittier 2013).

P. Lorkowski (✉) · T. Brinkhoff
Institute for Applied Photogrammetry and Geoinformatics (IAPG),
Jade University of Applied Sciences Wilhelmshaven/Oldenburg/Elsfleth,
Ofener Str. 16/19, D-26121 Oldenburg, Germany
e-mail: peter.lorkowski@jade-hs.de

T. Brinkhoff
e-mail: thomas.brinkhoff@jade-hs.de

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_9

One obstacle in this field is that sensor measurements tend to be arbitrarily distributed in space and/or time. Therefore, thinking in snapshot models derived from synchronous observations is insufficient. A more flexible approach is needed that complies with the structure of distributed single observations. Two major problems have to be solved within such a model:

1. ***Spatio-temporal interpolation*** between discrete measurements to provide distinct values for any unobserved point in space-time; such interpolations are needed for further analysis or applications (e.g., a continuous real-time map of the phenomenon).
2. ***Approximation or compression*** of given scenarios providing a reasonable compromise between accuracy and data volume.

Geostatistics provide sophisticated and widely proven methodologies for interpolation in a spatio-temporal context. The main idea is that measured values tend to be similar when proximate in space and time (Cressie 2011).

Therefore, given a limited set of measurements in space and time, for each non-sampled point within the (spatio-temporal) area of investigation a value can be estimated. This is done by a linear combination of the given measurements. Their weighting is performed in accordance with their spatio-temporal proximity to the point to be calculated. A deterministic method for such an approach is inverse distance weighting (IDW, see Whittier et al. 2013). For kriging, the dependence between distance and weight is expressed by the covariance function. Depending on the phenomenon, different types of covariance functions with different parameter settings are appropriate.

Unlike other interpolation methods, kriging also provides an estimation of uncertainty (or confidence respectively) for each calculated value by the associated kriging variance. When applied to a grid, it produces continuous maps of both: the values and the variances. The uncertainty estimation or deviation map (we use these terms synonymously) provides essential information whenever reliability is important. This is especially the case when dealing with inhomogeneous measurement data. Besides, the deviation map associated with each kriging calculation opens up promising approaches for filtering, real-time processing and archiving sensor data.

Given all the advantages above, the main disadvantage of kriging compared to other interpolation methods is its computational complexity of $O(n^3)$, which makes it hardly applicable for large datasets. This problem is of increasing importance because more sensor data can be expected in near future by more and cheaper sensors organized in sensor networks.

In order to deal with such large datasets, we propose a strategy that divides the large model into sub-models that are calculated separately before merged. For both, calculating and merging, we make use of the kriging method. For environmental real-time monitoring, the solution is embedded into a higher-level architectural pattern.

The remainder of the article is structured as follows: The next section describes the scope of problems for which the suggested method can be applied. Works

related to those problems are listed in Sect. 3. Kriging and its core element, the covariance function, are revisited in Sect. 4 and evaluated in respect to the proposed approach. Based on this discussion, our specific sequential approach is introduced in Sect. 5. The results of first evaluative simulations are presented in Sect. 6, conclusions and future work can be found in Sect. 7.

2 The Problem Domain

When thinking about environmental monitoring systems, it is useful to envision a machine that collects all information available for generating the most probable model of the environment. This model can be situated in the past, present or even future. It has to be in accordance with the available data as well as with the knowledge of the physical processes generating it (Jaynes 2003, p. 7ff). The quality and predictive power of such models has increased in the past. But no matter how elaborate, there will always remain limits:

- limited accuracy and density of measurements,
- limited knowledge of the physical processes involved,
- limited computational power for model calculation, and
- principally unpredictable factors like chaotic systems (e.g. weather) or social impacts (e.g., commuter traffic).

Nevertheless, within those limitations a reasonable estimation of unobserved regions is possible and useful, often even crucial. The field of environmental modeling has extensively been investigated. Principally, environmental phenomena can be modeled deterministically or stochastically (Cressie 2011).

Deterministic models are limited in their predictive power. Nevertheless, they are necessary for describing systematic effects like periodicity or trends. Statistics come into play whenever uncertainty has to be handled. In a spatial context, the method of kriging has developed "to be synonymous with 'optimal prediction'" (Cressie 1990). By using complex covariance functions, even systematic effects like periodicity can be incorporated into a stochastic model (Osborne et al. 2012).

Besides those improvements in modeling and processing, the following factors have massively increased the amount and accessibility of sensor data:

- Sensors are becoming increasingly efficient, cheap, light and mobile.
- Improving communication infrastructure allows ubiquitous data transfer.
- Increasing computational power allows complex model calculation in (near) real time.

With those possibilities, also the demands for (near) real-time monitoring systems are increasing. The ever growing awareness of environmental impacts on our lives is a strong driver for those technologies. As sensors become more frequent and connected via networks, the demand for an appropriate processing of captured data will also grow (Wikle 2003).

When dealing with continuous environmental phenomena like air pollution, one central challenge is to fill the spatio-temporal gap between the samples measured and the knowledge required. For instance, a continuous data stream provides ozone measurements by stationary and mobile sensors. For estimating the air pollution load of an urban district, it is necessary to interpolate spatio-temporally between the available samples. In addition to the calculated pollution values, an estimation of their uncertainty or prediction error is at least useful, if not indispensable.

In our research, we assume a continuous phenomenon that is observed by discrete samples, randomly distributed in space and time. From these samples, a continuous field of arbitrary resolution can be inferred using geostatistics.

The difference between the observed field and the inferred field depends on the density (and quality) of samples and the quality of the inference model.

Assuming numerous samples and using kriging because of its particular capabilities leads to the dilemma of high computational workload. Within this scope, we suggest a kriging-based method using sequential sub-models in order to reduce computing complexity and supporting real-time monitoring for massive data streams.

As a strategic orientation, we envision a sensor data stream processing environment that our module is embedded into. It adapts to data throughput, available computational power and requirements like accuracy, actuality and data volume. A sketch of this environment is depicted in Fig. 1 and described below.

While processing a continuous data stream, e.g. provided by web services of the OGC sensor web enablement (SWE) (Botts et al. 2007), the system consecutively updates a value map and its associated deviation map. This is done by a linear combination of each new generated map (derived from the latest set of samples) with the predecessor map. Because of the continuous character of both, the value map and the deviation map, the resulting model integrates new observations seamlessly. The system is called Data Stream Engine (DSE) in Fig. 1. It also includes an adaptive filter which is updated regularly and reduces data throughput when necessary. It prefers samples from regions yet poorly observed according to the actual deviation map.

Based on web services, the DSE provides interfaces for both: interactive web mapping and automated monitoring. For the latter, critical states can be defined and checked against the current map regularly. Those definitions can refer to values (e.g. for alert after exceeded threshold), deviations (more measurements necessary) or both combined (high risk of exceeded threshold).

For queries on historical data and long-term analytics, an archive containing data compressed by approximation is maintained alongside with the real-time services. When queried, it is decompressed and provided as usual map.

The methodology introduced in the next sections is in principle designed to support the functionalities of the environment sketched above.

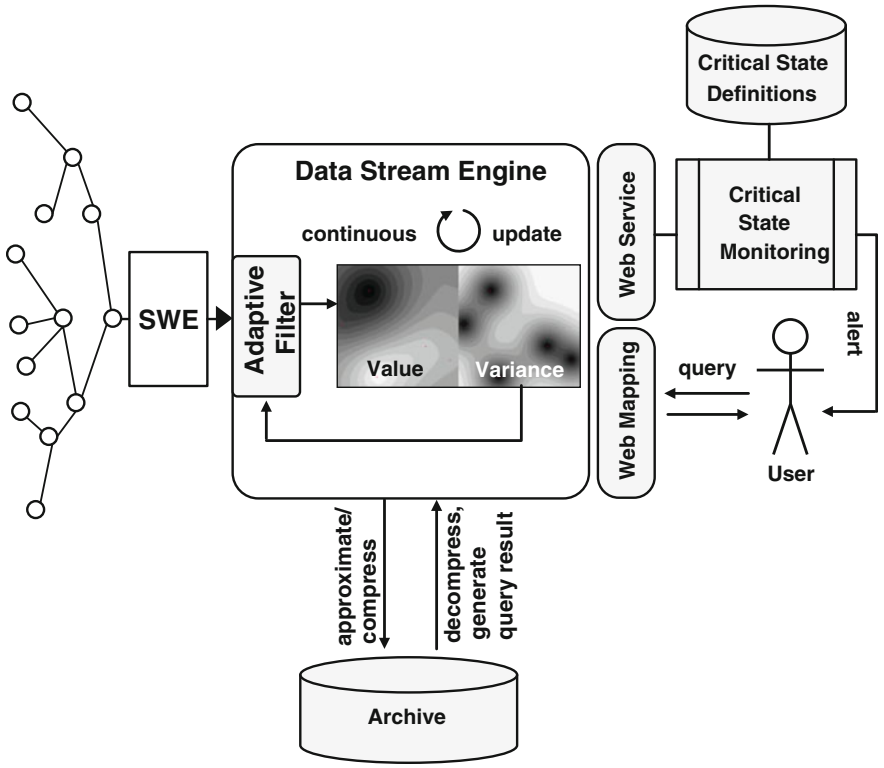


Fig. 1 Architecture of an environment that processes, visualises, monitors and archives sensor data streams

3 Related Work

Environmental monitoring of continuous phenomena deals with many issues, of which some are:

- deterministic physical models,
- sensor models (A/D conversion),
- signal processing,
- sensor communication/sensor web enablement (SWE),
- data processing (including complex event processing (CEP) and geostatistics), and
- visualization.

In the following, we will focus on data processing assuming high data rates combined with real-time requirements. We use the method of kriging because of its statistical validity and the estimation of uncertainty for each calculated point. While the field of geostatistics is very well established (Cressie 2011; Webster and Oliver

2007; Wackernagel 2003; Armstrong 1998), its application to real-time monitoring scenarios is relatively new. Some of the work done in this area is introduced below.

Whittier et al. (2013) suggest a new design of a Data Stream Engine (DSE) that is based on k Nearest Neighbors (kNN) and spatio-temporal inverse distance weighting (IDW). It uses main-memory indexing techniques to address the problem of real-time monitoring of massive sensor measurements. In contrast to this approach, we want to avoid a sub-model based on a fixed sized temporal interval. By merging sub-models continuously, we also consider old observations if no better information available. This might be especially important when observations are inhomogeneously distributed in space and time.

Appice et al. (2014) inspect trend clusters in data streams and discuss techniques to summarize, interpolate and survey environmental sensor data. Since one main application is the detection of outliers within a rather low dynamic phenomenon (solar radiation), the approach allows a coarse approximation by clusters of similar values. For our purpose, a smooth representation of each state is desirable.

Walkowski (2010) uses the kriging variance to estimate a future information deficit. In a simulated chemical disaster scenario, mobile geosensors are placed in a way that optimizes the prediction of the pollutant distribution. Instead of optimizing the observation procedure itself, we exploit the kriging variance in order to achieve efficient continuous model generation from massive and inhomogeneous data.

Katzfuss and Cressie (2011) decompose a spatial process into a large-scale trend and a small-scale variation to cope with about a million of observations. This solution is an option for optimizing very large models, but is not helpful for our sequential approach with its real-time specific demands.

Osborne et al. (2012) introduce a complex model of a gaussian process (synonym for kriging) that incorporates many factors like periodicity, measurement noise, delays and even sensor failures. Similar to our work, sequential updates and the exploitation of previous calculations are performed, but here on a matrix algebra basis. It uses kriging with complex covariance functions to model periodicity, delay, noise and drifts, but does not consider moving sensors.

4 Kriging

Because of its statistically founded methodology and uncertainty estimation (kriging variance), kriging was chosen as basis for our sequential approach. Herein, we divide a large set of observations into several subsets or sub-models, which are computed separately. Those sub-models are merged into a whole model again by applying policies which consider the confidence decay of observations with increasing spatio-temporal distance. For comprehensive understanding, we give an overview of the method of kriging and the associated spatio-temporal semivariances and covariance functions and discuss them in respect to the requirements of our approach.

Dynamic continuous phenomena tend to be stationary in space and time (Cressie 2011). For estimating the optimal value Y for an arbitrary position in space and time (s, t), a given set of samples Z(s, t) is compiled by linear combination. The relative weight of each sample depends on its spatial and temporal distance to the point to be estimated. Thus, spatially and temporally proximate samples have a stronger influence on the estimation than further ones. This reflects the continuous character of the phenomena discussed here.

Those phenomena tend to behave differently in space and time, which can be expressed by a spatio-temporal anisotropy. Instead of using $\mathbb{R}^3$ as position index, space and time are modeled separately as $\mathbb{R}^2 \times \mathrm{T}$. Thus, locally stable but temporally dynamic phenomena like air pressure, or locally dynamic but temporally stable phenomena like soil contamination can be modeled appropriately. Spatial anisotropy caused by physical phenomena (slopes, wind) will not be treated here; see (Cressie 2011, p. 12; Romanowicz 2005, p. 765) for further information.

For kriging, the central control element is the covariance function. It represents the stochastic assumption about the variances of the sample values depending on their distance or other factors; see (Cressie 1993) for a comprehensive study.

4.1 Spatial Covariance Function

Spatial autocorrelation can be described by the variogram. Given a dataset, for each possible pair of n samples, a distance d and its corresponding variogram $\gamma = ½ (z1 - z2)^2$ can be computed. (Because of the factor of ½, this indicator and its associated diagram are also called *semivariogram*. We will use the shorter term *variogram* here.) So, for n points, with $(n^2 - n)/2$ pairs and their corresponding parameters d and γ, the spatial correlation is empirically described and can be best interpreted when scatter plotted (see Fig. 2). As can be seen, the (semi-)variance increases with increasing distance. More systematically, this variance can be described by the theoretical variogram, which is a function of distance d and represents a trend line of the points (dashed line in Fig. 2).

Alternatively to the variance, the spatially dependent correlation of point values can be represented by the covariance function (solid line in Fig. 2). It is the mirror image of the theoretical variogram about a line parallel to the abscissa (see Webster and Oliver 2007, p. 55f). It expresses the autocorrelation or self-correlation of the observed variable which decays with increasing distance.

From the many possible covariance functions and their associated variogram models (Cressie 1993; Armstrong 1998; Wackernagel 2003; Webster and Oliver 2007), the exponential covariance function is chosen here because it is commonly used (Webster and Oliver 2007, p. 88ff) and, unlike the spherical covariance function, asymptotically approaches but never reaches zero. For sub-models with only few points this is an important issue, since even far apart regions should be influenced by the samples.

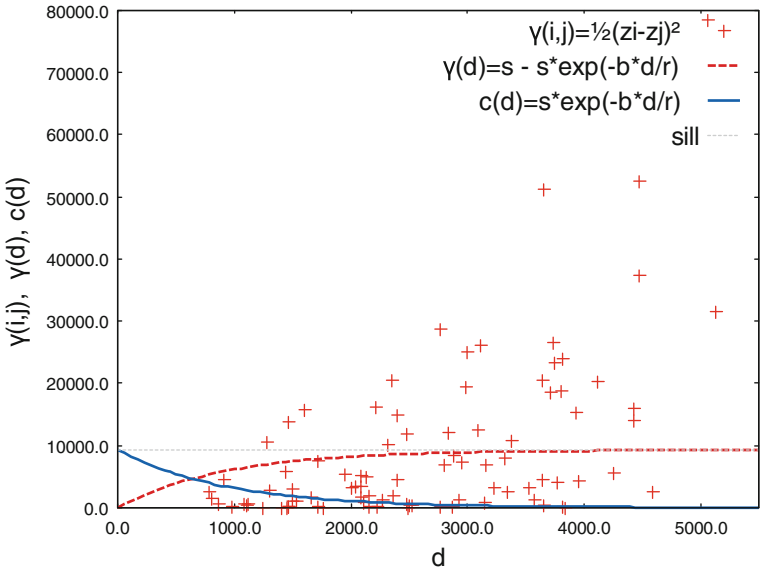


Fig. 2 Geostatistical properties of a sample set: empirical variogram (*scatterplot*), theoretical variogram (*dashed graph*), covariance function (*solid graph*) and sill (*dashed horizontal line*)

The parameters of both the variogram and covariance function are *sill* and *range* (*s* and *r* in Fig. 2), where *sill* is the horizontal asymptote and *range* marks the point where *sill* is nearly (here: 95 %, see below) reached by the theoretical variogram graph. The parameter *sill* is given by the variance of all values of the set (see Armstrong 1998, p. 21), while *range* can be estimated by linear regression or by more sophisticated methods (Cressie 1985). The third parameter (*b* in Fig. 2) determines the decrease rate of the covariance function. When set to 3, the function has decreased by 95 % when the distance has reached *range* (see Wackernagel 2003, p. 57ff).

The covariance function $c(d)$ is an essential element of the kriging calculation. Based on this function, for each unobserved position an unbiased estimation of both the value and its corresponding variance is performed. When applied area-wise, this variance can be used to define a weight when two models of the same area but different samples have to be merged. Thus, each point value in the merged model is influenced by the corresponding point values of the source models according to their variance. We apply this principle to merge separate models in a sequence.

4.2 Temporal Covariance Function

For the temporal correlation, we choose a slightly different approach. The covariance function is shown in Fig. 3 and implements the half-life principle of temporal

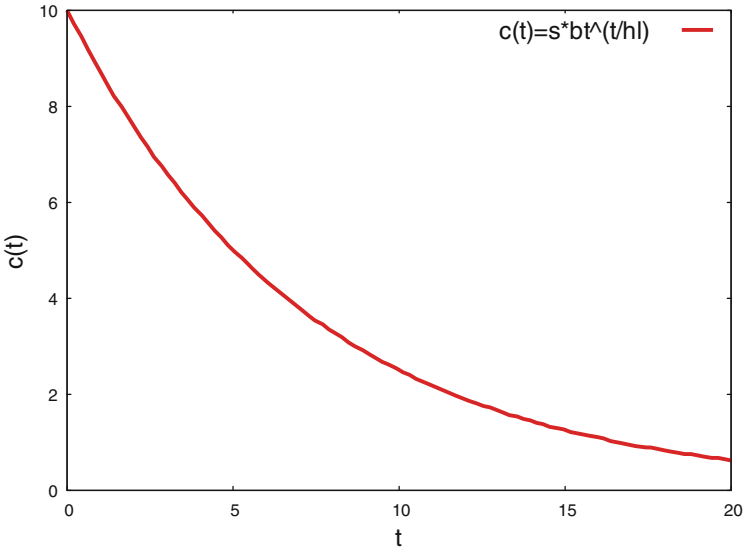


Fig. 3 Temporal covariance function representing the informational decay over time

decay. As for the spatial covariance, the parameters of this function can also be estimated by empirical data, namely time series at fixed spatial positions.

4.3 Spatio-temporal Covariance Function

In a real-time environmental monitoring scenario, space and time have to be considered when generating appropriate weights for each sample and thus estimating the value of a non-sampled position. Given the spatial and temporal covariance functions above, the total "correlation decay" based on spatial *and* temporal distance can be calculated. This is done by multiplicative combination of the two functions resulting in a 3-dimensional model (see Fig. 4). Given this function, for any spatio-temporal distance a distinct covariance is defined, while the spatial dimension is handled differently from the temporal dimension; see also (Cressie 2011, p. 302ff; Walkowski 2010, p. 71). Spatial anisotropy, as already mentioned, is not considered here.

This spatio-temporal covariance function can be applied whenever samples are time-stamped. Thus, besides the spatial information decay also the temporal information decay can be considered. When merging sub-models (see Sect. 5), due to temporal decay their weighting can be determined by the reference timestamp (newest sample) of the model. Therefore, only the temporal component of the function is required (see Fig. 3).

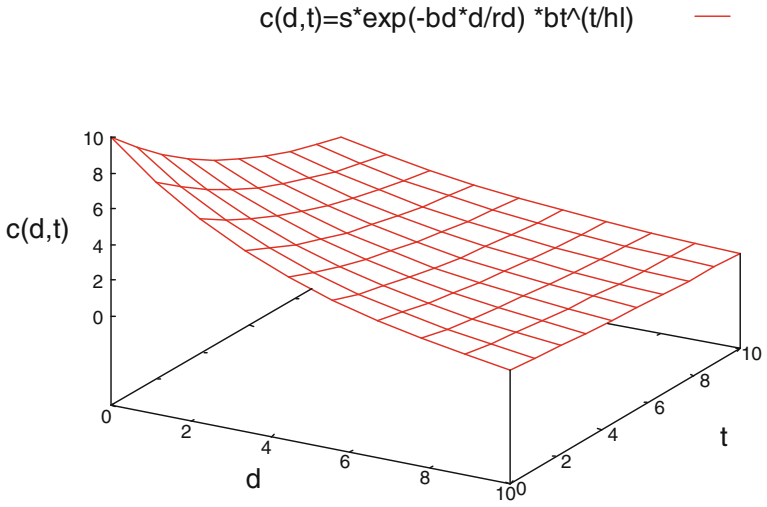


Fig. 4 Spatio-temporal covariance function $c(d, t)$ representing the decay of correlation as a function of spatial distance (d) and temporal distance (t)

Apart from the considerations above, no further examinations of the spatio-temporal correlation had been undertaken yet. But to justify the broader perspective of a system operating in real time, it is necessary to consider the general concept.

With the covariance functions above, a statistical description of continuous fields in a spatio-temporal context is given, when all systematical influences (e.g. drift, periodicity) are removed. Parameters like *sill* and *range* can either be estimated a priori (Bayesian Hierarchical Model, BHM) or derived from the data (Empirical Hierarchical Model, EHM) (Cressie 2011).

Based on the stochastic description of spatio-temporal processes expressed through covariance functions, estimations for any point can be done by a linear combination of given samples. Alongside with the value estimation, kriging provides also an estimation of variance (or deviation resp.) for each point calculation. This represents the basis for the weighted merging procedure described in the next sections. When applied area-wise to a grid, both values form a continuous map for visual interpretation (see Fig. 5).

5 The Sequential Algorithm

Notwithstanding the advantages mentioned above, kriging comes along with a computational complexity of $O(n^3)$ (Osborne et al. 2012; Barillec et al. 2011), n being the number of samples. Considering this fact in the context of massive data load in combination with (near) real-time requirements, this can become a severe limitation of the method. Hence, when sticking to the essential advantages of

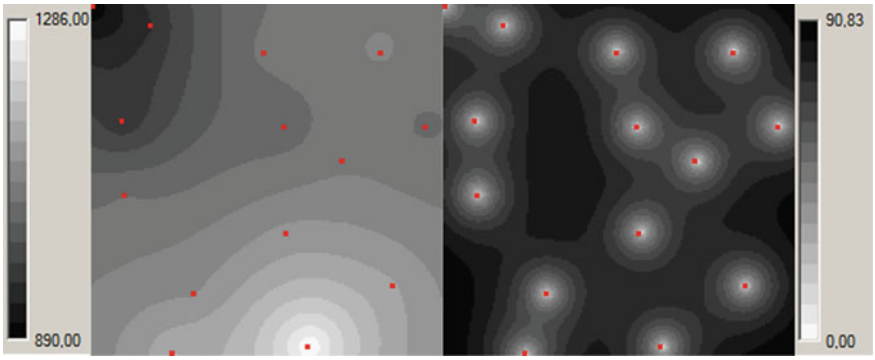


Fig. 5 Kriging result with value map and corresponding deviation map

kriging, an appropriate strategy to cope with the computational complexity problem is required.

For overcoming this problem, we separate the whole set of observations into s subsets for which the kriging method is applied separately. The resulting sub-model grids cover the same area as the master model that contains all points. To consider all measurements in the final model, the sub-models are sequentially merged with their predecessor, as shown in Fig. 6. The loss of accuracy of this approach is quantified by the Root Mean Square Error (RMSE) against the master model.

In a spatio-temporal context, the segmentation should be performed in respect of the order of timestamps, thus representing temporal intervals per sub-model. This applies also to real-time environments where subsequent models would be created continuously. For a pure spatial model, the subsets of points can be generated randomly. Here, the order of sub-models does not represent the temporal dynamism of the phenomenon, but rather a utilisation level of information with associated estimated accuracy. This is also the case for our experiments introduced below.

The segmentation and associated sequential calculation limits the potential complexity of $O(n^3)$ to the size of each subset s. This can be set as a constant, but could also be dynamically adaptive to the data rate. In any case, there should be an upper bound for the size of sub-models to limit the computing complexity.

While doing so, the merge procedure itself can be costly, but grows only linearly with n and can easily be parallelized. Thus, it is not principally critical for massive data.

The general characteristic of the computational complexity of this approach is compared to the one of the master model calculation in Fig. 7. As can be seen from the formula depicted in the lower part of the figure, the reduction of complexity is achieved by removing n (except $n \bmod s$, which is uncritical) from cubed terms.

Having averted the $O(n^3)$ complexity by separating into subsets, an appropriate strategy is needed for the merging procedure. It should produce results close to the master model that contains all samples. In order to achieve this objective, the confidence estimate (derived from the covariance) of both models is considered in

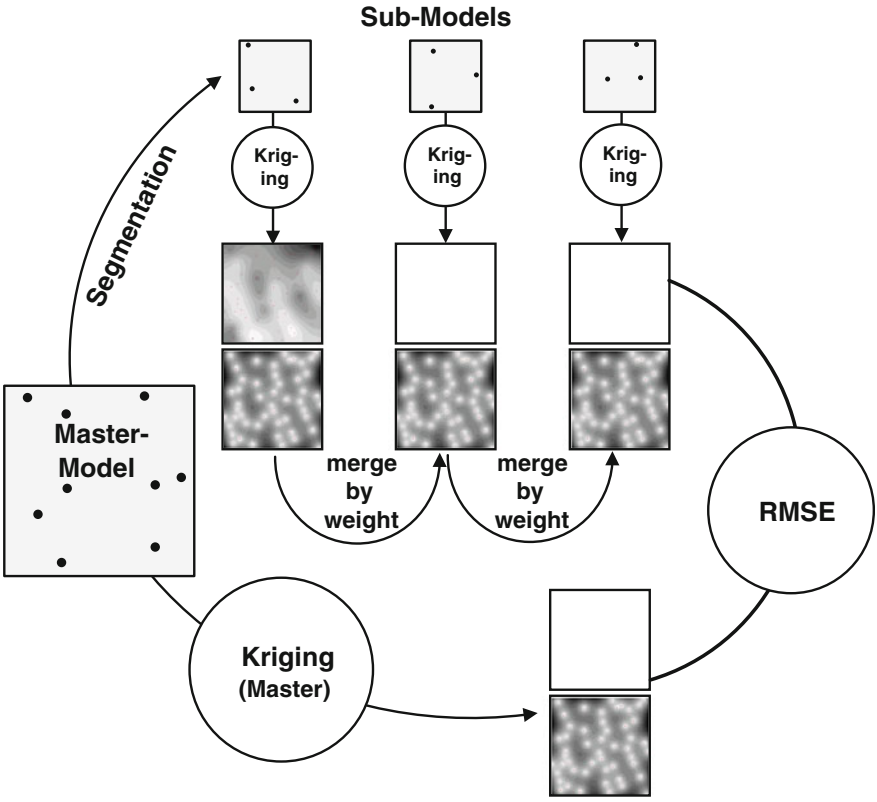


Fig. 6 Sequential calculation scheme

each merge step for defining a weighting factor. This is done for every sampled position in a grid, resulting in a new grid model or map.

As can be seen in Fig. 8, alongside with the value map also a new weight map is generated by each merge procedure, representing a continuous confidence field of the new generated model. It is created by adding the weights of both models before normalizing all weights to a predefined interval. When this interval is set to [0.1; 1.0] for example, also low weighted areas can remain in the model for many sequences, unless overwritten by new observations. When the temporal difference between the models has to be considered, the older weight map is multiplied by the temporal decay factor before added. This factor is derived when the time difference between the models is entered into the temporal covariance function.

Assuming this merging procedure, temporally old or spatially isolated observations can keep their influence over many merging steps. This is especially helpful when no better observations are available. Nevertheless, the growing uncertainty of this estimation would always be expressed alongside by the associated confidence map.

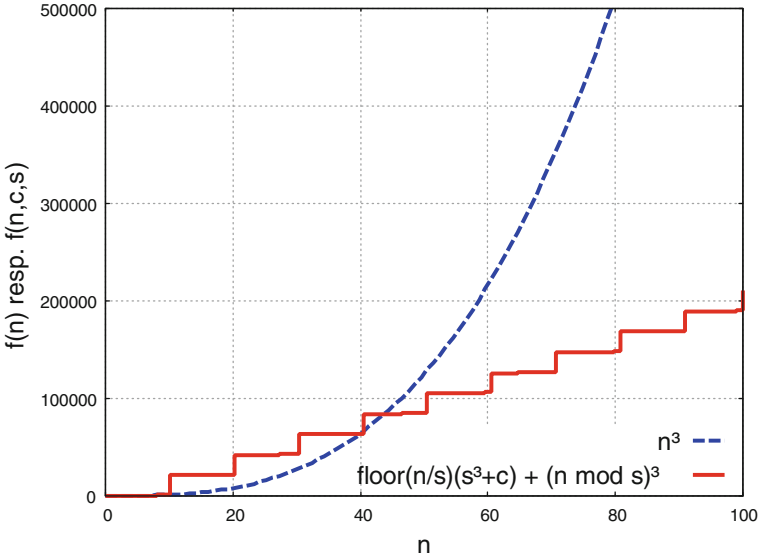


Fig. 7 Theoretical complexity of master model calculation (*dashed line*) versus the sequential method (*solid line*); *n* all samples, *s* size of sub-model, *c* merging effort

Figure 8 illustrates the merge procedure by showing value maps with their corresponding confidence or weight maps. Merging the two sub-models means to create a new value map based on both source maps weighted by their corresponding weight maps. On the other hand, a new weight map is also created as a combination of the input weight maps, now describing the confidence of the new merged model as continuous field.

Ideally, the resulting map would be identical to the master model which is based on all observations. Since this is principally impossible due to the loss of correlational information by sequencing, the aim must be to compute a map as similar as possible by configuring the merge process.

Apart from this loss of calculative accuracy, the strategy of sequencing comes along with several advantages. On the one hand, it can be used to calculate large datasets with less computational effort. In principle, kriging properties like unbiased, smooth interpolation and uncertainty estimation are kept.

On the other hand, given a continuous sensor data stream, this approach can integrate new measurements seamlessly into the previous model at flexible update rates. Providing also the updated weight map with each new integrated sub-model, an adaptive filter can be constructed that ignores additional measurements in already well-sampled regions. Such filter is especially useful in the case of high data rates.

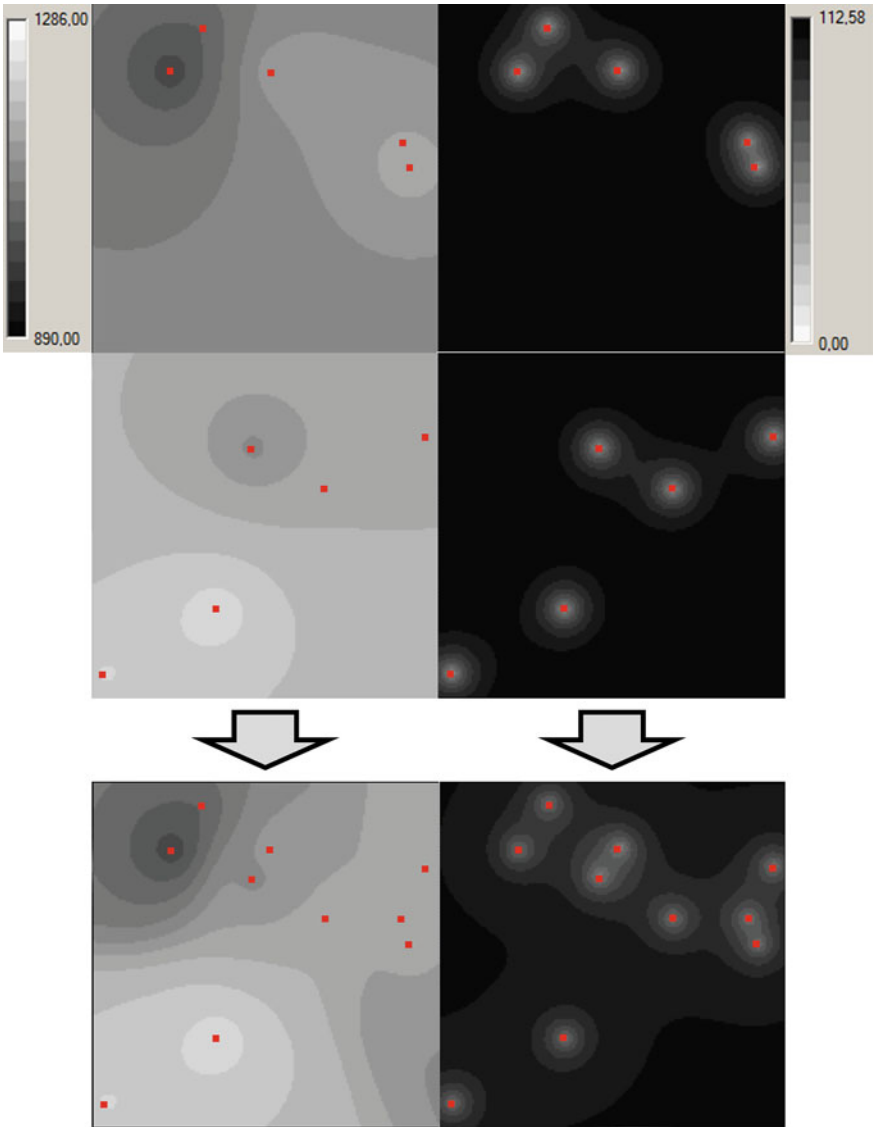


Fig. 8 Merging of models by using weight maps

6 Evaluation

In order to prove the feasibility of the presented approach, but also to reveal its effects on accuracy, a simulation with appropriate indicators is performed. We chose a synthetic grid data model for this. It is derived by kriging over 14 rain

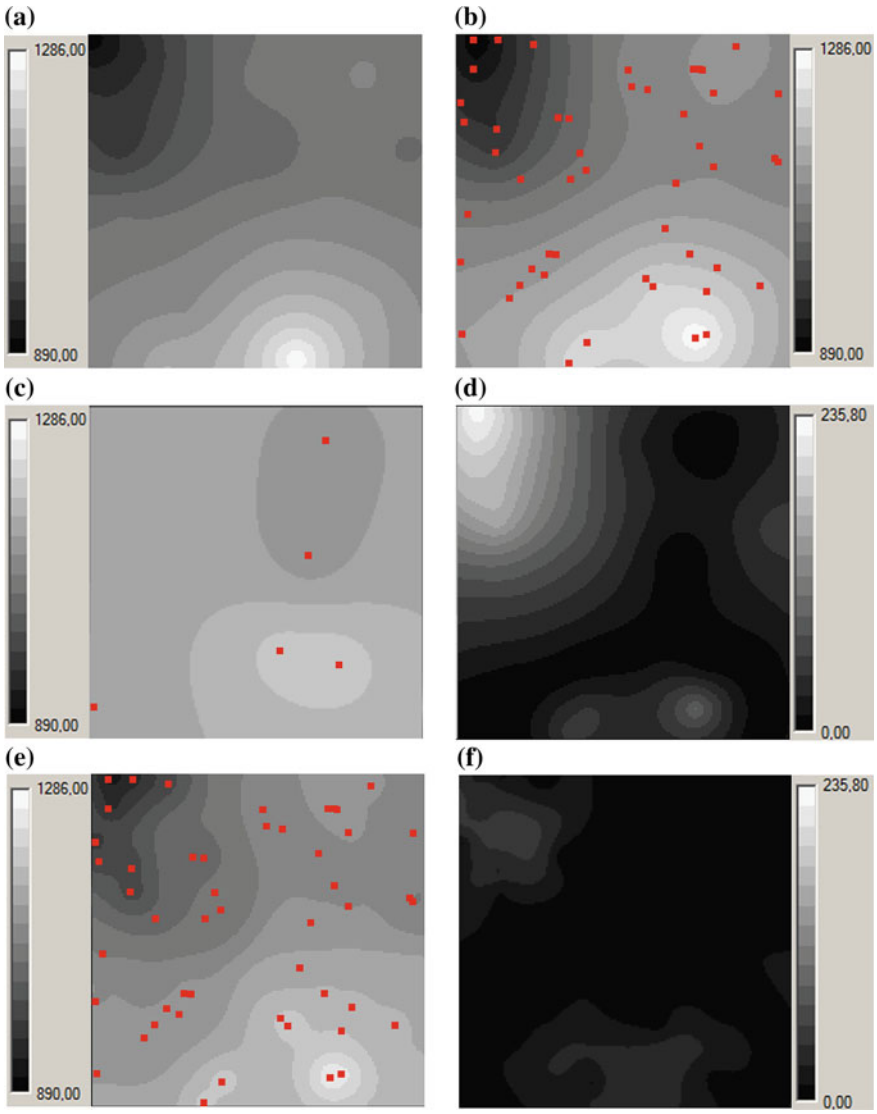


Fig. 9 Evaluation of sequential method: reference model (**a**); random points and corresponding model derived from those points (**b**); first subset of random points (**c**) with corresponding difference map (**d**) towards reference model (**a**); sequentially updated model of all subsets (**e**) with resulting difference map (**f**) towards the reference (**a**)

gauge stations and depicted in Fig. 9a. At this stage, we ignore temporal dynamism in order to keep it out as a factor for differences (RMSE) between the reference model and the sequential approach.

The program used for calculation is implemented for the .NET framework as managed C# code, partially using multithreading. It was also tested against the free and open source framework *Mono*,[1] which would allow running the system on other operating systems than Microsoft Windows.

6.1 Test Design

In the simulation scenario, the continuous grid model serves as reference (Fig. 9a). Random points are scattered over the model area, each assigned the value picked from the reference model at its position. Given this synthetic measurement set, a new model can be calculated by kriging (Fig. 9b).

The derived model (b) slightly differs from the reference model (a) due to interpolation uncertainty, but approximates it well when the number and distribution of samples are sufficient. Following the sequential strategy, subsets of all synthetic measurements are created and calculated sequentially in sub-models. For the first subset (Fig. 9c), the deviations to the reference model (a) are rather large and can be seen in the difference map (d). Calculating all subsets of the data and merging them successively by weight (see Fig. 8) leads to the final model (e), which also considered all the sample data, but unlike model (b) in a sequential manner. The difference map (f) expresses the discrepancy towards the reference caused by the sequential approach.

The overall discrepancy per model can be quantified by the root-mean-square error (RMSE) relative to the reference model (a). In the following, this value is used to indicate the quality of models.

6.2 Results

In Fig. 10, the computing time is plotted against the RMSE relative to the reference model for both: the complete model calculation (square) and the sequential method (connected dots). Randomized sets of points (100, 200, 300 and 400) were subdivided into subsets or sub-models of 10 points each.

As can be seen from the results, the sequential method has a lower accuracy in total, but provides a coarse result almost immediately. Within each plotted scenario, the RMSE tends to decrease when following the sequence.

The $O(n^3)$-effect of the conventional calculation becomes obvious when comparing its total computing time to the one of the sequential approach for large n.

[1] http://www.mono-project.com.

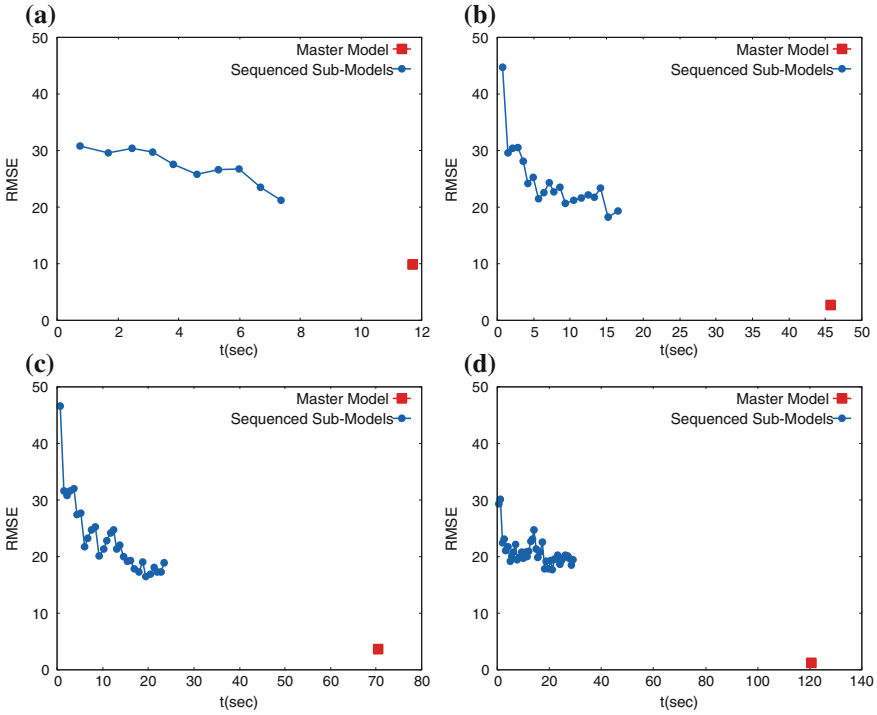


Fig. 10 Performance comparison between master model and sequenced calculation: **a** 100 samples, **b** 200 samples, **c** 300 samples, **d** 400 samples; subdivision is done in a way that sub-models contain 10 samples each

Since the samples are distributed randomly over the reference model, the results also tend to scatter when the scenario calculation is repeated, but remain similar in essence.

The tests introduced here are designed to explore the general behavior of our approach. It converges to a saturation value and for large models clearly outperforms the conventional method in computing time.

7 Conclusions

In this paper we addressed the problem of modeling continuous environmental phenomena from discrete samples arbitrarily distributed in space and time. Our method is based on kriging as a statistically consistent methodology. We proposed a novel approach to defuse the computational complexity of kriging for large datasets while retaining its capabilities. It works by dividing the dataset into sub-models, computing them separately and merging them based on their kriging variance.

The approach reduces total computing time for large datasets, provides coarse models immediately and defines a rule for seamless fusion of partial models. For real-time monitoring systems fed by a continuous data stream, this approach provides fast responsiveness and can adapt to data load and available resources. In a first experimental evaluation we examined the inevitable loss of information caused by the approximation and quantified by the RMSE (Root Mean Square Error).

In addition, the results of this evaluation show that the accuracy of the sequential sub-models converges towards an error level that appears reasonable for real-time applications. Nevertheless, there is still room for improvements in the way the sub-models are merged. Some deeper study of this aspect might reveal better strategies here.

The next steps will be systematic tests with different reference data, different parameter settings applied to different merging algorithms. We expect significant hints for refinement of the merging algorithm.

Furthermore, in order to use our approach in a realistic scenario with mobile sensors, we currently implement a simulation of moving objects. It is based on the bus schedules of the City of Oldenburg applied to OpenStreetMap data, thus simulating the spatio-temporal distribution of the bus fleet. Since such data will be inhomogeneous (only on bus routes; dense in the centre, scarce in the outskirts), we expect indications about the practical feasibility of our approach. Applying the method to genuinely measured sensor data will be the next step.

References

Appice, A., Ciampi, A., Fumarola, F., & Malerba, D. (2014). *Data mining techniques in sensor networks: Summarization, interpolation and surveillance*. London: Springer.

Armstrong, M. (1998). *Basic linear geostatistics*. Berlin: Springer.

Barillec, R., Ingram, B., Cornford, D., & Csató, L. (2011). Projected Sequential Gaussian Processes: a C++ tool for interpolation of large data sets with heterogeneous noise. *Computers and Geosciences, 37* (2011), 295–309.

Botts, M., Percivall, G., Reed, C., & Davidson, J. (2007). OGC® sensor web enablement: Overview and high level architecture. Open geospatial consortium. Online document: http://portal.opengeospatial.org/files/?artifact_id=25562.

Cressie, N. A. C. (1985). Fitting variogram models by weighted least squares. *Mathematical Geology, 17*(5), 1985.

Cressie, N. A. C. (1990). The origins of kriging. *Mathematical Geology, 22*(3), 1990.

Cressie, N. A. C. (1993). *Statistics for spatial data*. New York: Wiley.

Cressie, N. A. C., & Wikle, C. K. (2011). *Statistics for spatio-temporal data*. Hoboken: Wiley.

Jaynes, E. T. (2003). *Probability theory*. Cambridge: Cambridge University Press.

Katzfuss, M., & Cressie, N. A. C. (2011). *Tutorial on fixed rank kriging (FRK) of CO_2 data*. Technical Report No. 858. Department of Statistics, The Ohio State University.

Osborne, M. A., Roberts, S. J., Rogers, A., & Jennings, I. R. (2012). Real-time information processing of environmental sensor network data using bayesian gaussian processes. *ACM Transactions on Sensor Networks*, *9*(1), 1.

Romanowicz, R., Young, P., Brown, P., & Diggle, P. (2005). *A recursive estimation approach to the spatio-temporal analysis and modeling of air quality data*. Amsterdam: Elsevier.

Wackernagel, H. (2003). *Multivariate geostatistics: An introduction with applications*. Berlin: Springer.

Walkowski, A. C. (2010). *Modellbasierte Optimierung mobiler Geosensornetzwerke für raumzeitvariante Phänomene*. Heidelberg: AKA Verlag.

Webster, R., & Oliver, M. A. (2007). *Geostatistics for environmental scientists (statistics in practice)*. West Sussex: Wiley.

Whittier, J. C., Nittel, S., Plummer, M. A., & Liang, Q. (2013). Towards window stream queries over continuous phenomena. In: *4th ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS), Orlando*.

Wikle, C. K. (2003). Hierarchical models in environmental science. *International Statistical Review, 71*(2), 181–199.

Statistical Learning Approach for Wind Speed Distribution Mapping: The UK as a Case Study

Fabio Veronesi, Stefano Grassi, Martin Raubal and Lorenz Hurni

Abstract Wind resource assessment is fundamental when selecting a site for wind energy projects. Wind speed is influenced by a plethora of environmental factors and understanding its spatial variability is key for determining the economic viability of a site. Deterministic estimation methods, which are based on physics, represent the industry standard in wind-speed mapping. Over the years, these methods have proven capable of estimating wind speed with a relatively high accuracy. However measuring stations, which provide the starting data for all wind speed estimations, are often located at a distance from each other, in some cases, tens of kilometres or more. This adds an unavoidable level of uncertainty to the estimates, which deterministic methods fail to take into account. For this reason, even though there are ways of determining the overall uncertainty of the estimation, e.g. cross-validation, deterministic methods do not provide means of assessing the site-specific uncertainty. This paper introduces a statistical method for estimating wind speed, based on spatial statistics. In particular, we present a statistical learning approach, based on ensembles of regression trees, to estimate both the wind distribution in specific locations and to assess the site-specific uncertainty.

Keywords Wind speed · Statistical learning · Geostatistics · Weibull distribution · Random forest

1 Introduction

Wind energy can play a key role in reducing the level of CO_2 emissions required to avoid the worst effects of climate change. The United Kingdom has, for instance, pledged to reduce its carbon emissions by 80 %, compared to the 1990 baseline, by

F. Veronesi (✉) · S. Grassi · M. Raubal · L. Hurni
Institute of Cartography and Geoinformation, ETH Zürich
Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland
e-mail: fveronesi@ethz.ch

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_10

2050 (http://www.legislation.gov.uk/ukpga/2008/27/contents). With the depletion of conventional energy sources and the increase in global warming, renewable energy sources (RES) have attracted the interest of investors. Among the RES, wind energy has had a growth of 27 % over the last five years for a total installed capacity of 230 GW at the end of 2011 (REN21 2012) with an overall turnover of 50 billion euros (Pitteloud 2012).

When selecting a site for a wind energy project, a wind resource assessment plays a fundamental role, not only for practical purposes, but also for investment purposes. Meteorological stations collect climate data, but they are sparsely located and therefore do not provide the full data coverage necessary for an optimal placement of wind farms. In order to obtain wind speed data in unknown locations, we require a way to model the wind resource. The use of wind meso-scale maps with a resolution of a few kilometres is being used more and more in feasibility studies for wind energy projects in order to identify suitable sites. Several methods and approaches have been developed in recent decades to assess the mean wind speed over large regions (Landberg et al. 2003). These methods can be physical (deterministic) or statistical.

Physical models are methods that take data, such as topography, land-use, temperature and pressure, and solve models based on wind flow or fluid-dynamic equations. They can be divided by level of sophistication or complexity. Low complexity methods are linear wind-flow models, such as WAsP (2015, http://www.wasp.dk/), which are based on the theory of Jackson and Hunt (Jackson and Hunt 1975). More complex methods, albeit still linear, are the Reynolds-averaged Navier-Strokes model (RANS or CFD) (Snel 1998). These models assume steady-state flows and are therefore computationally relatively efficient and can be run on standard PCs. The next level of sophistication is occupied by non-hydrostatic weather prediction models (NWP), such as MM5 (2015, http://www.mmm.ucar.edu/mm5/overview.html) and the Weather Research and Forecasting Model (WRF 2015) (http://www.wrf-model.org/index.php). These models require a substantial amount of computing power that increases rapidly with finer resolutions.

In other studies, wind speed has been assessed using statistical and geostatistical algorithms (Beaucage et al. 2014; Cellura et al. 2008; Luo et al. 2008; Cellura et al. 2008). In this case, wind speed is estimated based on the spatial correlation of measured data with environmental predictors, such as topography and land-use, to provide additional information to the statistical model and thus increase its accuracy.

The main differences between these two methodologies are in the data requirements, their accuracy and the required computational time. Published research regarding the application of physical methods reports studies where, in general, only a very limited number of measuring stations are used for the investigation (a few examples are: Snel 1998; Meng et al. 2013; VanLuvanee 2009; Susumu et al. 2009). The flipside is that physical methods often require much more computer power and time to produce their estimates (Gasset et al. 2012). In contrast, statistical methods often require a substantial amount of observed data to be used for training (Foresti et al. 2011), but they tend to be more computationally

efficient. However, the statistical methods cited above also tended to be less accurate compared to the physical methods in our review. Linear physical methods present an increment of around 30 % in accuracy, while more complex ones are 70–80 % more accurate than the geostatistical methods used for wind speed estimations.

Another crucial issue that differentiate physical from statistical methods is the inability of the former to provide a measure of the site-specific uncertainty of the wind speed predictions (Huang et al. 2002). All estimations contain some errors and therefore cannot be exact. In addition, providing only the average error of the wind speed prediction does not tell much about its spatial distribution. However, if we are planning to use the wind speed map for selecting sites for wind farms, we must be able to express the site-specific accuracy of the estimations.

One of the main issues when selecting a site for a wind energy project is a spatial assessment over large regions with regard to the investment risk, which includes the wind resources available, the financial risk, and the uncertainties related to construction and operations (Pinson 2006). If one excludes the risks related to financial aspects, in addition to construction and maintenance failures, which are not addressed in this work, then an optimal pre-feasibility assessment of wind resources is critical and fundamental because it is subject to various uncertainties and can thus significantly impact the success of a project (Grassi et al. 2012).

Approaches to quantify the uncertainties of the wind energy output have been developed using different methodologies, such as Monte-Carlo simulation or Measure–Correlate–Predict methods (Gass et al. 2011). However, these approaches work by using local wind measurements, sometimes measured on the planned site by installing ad hoc measuring towers, and by calculating the propagation of the wind variability on the energy output. In this work, we provide a local measure of the wind variability that can be used for this type of analysis, thus avoiding the initial step of installing local measuring stations.

The main purpose of this research is to create a framework to support planners during the feasibility study in order to identify locations suitable for wind energy projects, without the need for additional time-consuming wind measurement campaigns. Additionally, we want to create a map that, in addition to the wind distribution, shows the spatial distribution of the prediction uncertainty. This would provide planners with more detailed information during the pre-feasibility study, so they can refine the estimate of the future energy production of selected sites and rank them as a function of the economic risk related to the wind resource uncertainties. In general, the map would provide a tool for a detailed spatial evaluation of the investment risk related to the level of exploitable wind sources.

In this research, we developed a method that uses statistical learning to spatially estimate the two parameters of the Weibull distribution at meso-scale resolution. These two parameters uniquely identify the local wind distribution over a given time period. By using them, we were able to estimate the local distribution of the wind speed at 1 km resolution across the United Kingdom. This potentially allows the computation of the probability that the wind maintains a speed between the cut-in and cut-off speeds, and therefore the amount of time that the wind resource is

economically exploitable. Moreover, we developed a technique for assessing the local uncertainty of the mean wind speed. This allows a quick and easy assessment of the local reliability of the map providing the probable range of variation of the wind resource.

2 Materials and Methods

2.1 Study Area and Dataset

This research was conducted in the United Kingdom over a total area of 244,119 km^2 at 1 km resolution. The study area with the location of the meteorological stations is shown in Fig. 1.

Wind speed distributions were obtained from 188 stations across the United Kingdom for a time period between 2009 and 2013. The data are part of the MIDAS Wind Data Archive and are freely available for research purposes from their Website (Kwon 2010). These data are referred to as training locations and visualised as red points in Fig. 1. This dataset is composed of long-term wind speed measurements taken at hourly intervals. We excluded data from the archive stations with less than six months of measurements.

Several covariates were used to perform the estimation. In particular, the Aster DEM (30 m resolution) provided by NASA (Met Office 2012) was used for elevation data and to create DEM derivatives, such as slope, aspect, and roughness [computed in SAGA GIS (Center 2011)]. The land-use raster data (100 m resolution) were provided by the CORINE project (Conrad 2007). In addition, we used raster maps at 5 km resolution from the MET Office (2006) with meteorological data, e.g. mean annual temperature, maximum temperature, minimum temperature, mean atmospheric pressure, air frost, cloud cover, rainfall, and relative humidity.

2.2 Weibull

Wind speed is usually measured and collected over years by meteorological stations and towers, which belong to the national meteorological services, and are placed at airports or in the proximity of a wind farm. The wind speed characteristics are usually (but not only) measured at 10 m above ground level. In order to assess whether the wind speed at a given location is economically viable, statistical analyses of the wind data are carried out to quantify its probability distribution. The probability distribution describes the likelihood that a given value will occur, therefore, the longer the data collection, the more reliable the probability distribution. For a wind field, this distribution is described by the Weibull distribution,

Fig. 1 Locations of MIDAS weather stations

which is a case of the generalised gamma distribution (Jenkins et al. 2008). The Weibull distribution is a two-parameter continuous probability distribution (Agarwal and Kalla 1996) that is defined by the following density function:

$$f(x;C;k) = \begin{cases} \frac{k}{C}\left(\frac{x}{C}\right)^{k-1} & e^{-(x/C)^k}, \quad x \geq 0 \\ 0, \quad x < 0 \end{cases} \tag{1}$$

where x is the wind speed (in m/s), k is the shape parameter and C is the scale parameter.

This function is generally used in the literature to describe wind distributions (Gass et al. 2011; Manwell et al. 2009; Akpinar and Akpinar 2005), and it can be described by only relying on two parameters: shape and scale. Thus, if continuous spatial estimates of k and C are available, the probability distribution of wind speed in any given location can be predicted.

Figure 2a, b present the influence of C and k on the density function. Generally speaking, a variation of the C parameter, keeping k constant, directly affects the mean wind speed that increases proportionally with an increase in C. In contrast, a variation of k, keeping C constant, produces a decrement in the dispersion of the measurements around their mean. These two parameters uniquely identify a Weibull distribution, and therefore can be used to estimate one in space. Figure 2c shows an example of a typical wind distribution with the fitted Weibull distribution.

2.3 Statistical Learning Approach

The local wind speed distribution is influenced by many environmental covariates, such as topography (Munteanu et al. 2008) and land-use (Schmidli et al. 2010; Rogers et al. 2005). Therefore, we selected the statistical learning class of algorithms (Ray et al. 2006) that can estimate the variables of interest for this research (in this case, the Weibull parameters) by finding relationships between them and environmental covariates, which are referred to as predictors.

Statistical learning is a branch of statistics aimed at modelling and understanding complex datasets (Kotsiantis et al. 2007). These algorithms can be divided into supervised and unsupervised estimators, the former is used in this research. Supervised algorithms require training to be able to model the observed process with the help of predictors. This way, the algorithm can define a set of rules that can then be used to estimate a variable in an unknown location where only the predictors are available.

Another crucial objective of this study is the estimation of site-specific uncertainty. To do this, we needed an algorithm capable of assessing its own accuracy. We selected Random Forest (RF) (James et al. 2013) as our mapping tool, which is based on ensembles of regression trees. These types of algorithms create regression trees by splitting the training dataset according to rules that minimise the residual sum of squares (RSS) between the measured data and the estimated data. Each split of the data is performed by testing different predictors in order to find the one split that minimises the RSS in the resulting subsets, or tree leaves (Breiman 2001). A regression tree can be viewed as a series of 'if-then' rules that are used to define

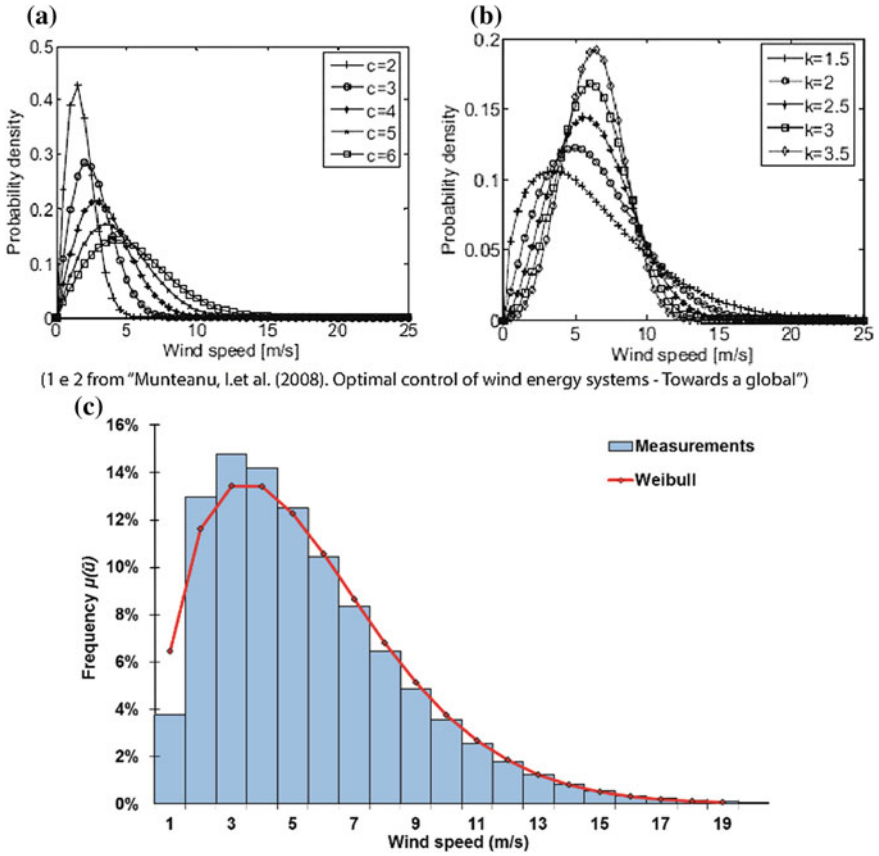


Fig. 2 **a** Weibull distributions as a function of C (constant k, *source* Munteanu et al. (Yim et al. 2007)); **b** Weibull distributions as a function of k (constant C, (Yim et al. 2007)); **c** Example of a fitted Weibull distribution (*red line*) plotted on top of a wind speed histogram (*blue bars*) with bins of 1 m/s

classes of probabilities. In a location where no wind data is available, the prediction is performed by running the predictors through the tree in order to define the most probable value for that particular location.

Regression trees have the advantage of being very easy to explain and interpret. The regression tree can be graphically displayed and this is a great advantage compared to other methods. However, regression trees generally have a lower predictive power than other approaches (Kotsiantis et al. 2007). RF solves this by using ensembles of trees, bagging and decorrelating the single trees.

Random Forest is based on ensembles of regression trees. Basically, instead of using the entire dataset to build one single regression tree, it uses the bootstrap to build numerous trees, starting from the same dataset. Bootstrapping is a statistical resampling method, which takes a dataset with n observations and resamples it

using replacements, meaning that an observation can occur more than once in bootstrap samples. This produces a series of samples, of length n, to which the algorithm can fit a regression tree. This means that RF can fit numerous regression trees to the same dataset; this procedure, technically referred to as bagging, reduces the variance of the method and therefore increases its accuracy (Kotsiantis et al. 2007).

Random Forest has another advantage compared to pure bagging, it subsets the predictors at each split of the tree thus decorrelating the ensemble. This is a strong advantage that leads to higher accuracy. The reason is simple: Suppose we have one predictor that is strongly correlated with wind speed. If we allow the algorithm to choose among all predictors, most or all of the trees will use this strong predictor. As a consequence, all the trees will be highly correlated and this does not lead to a substantial reduction of variance (Kotsiantis et al. 2007). Random Forest overcomes this problem to achieve even higher estimation accuracy.

Random Forest has been widely used in research for different purposes, such as digital soil mapping (e.g. Hansen et al. 1996; Grimm et al. 2008), ecology (e.g. Wiesmeier et al. 2011) and remote sensing (e.g. Cutler et al. 2007; Chan and Paelinckx 2008). Its popularity is due to the fact that it can generate reliable estimates and it is robust against noise in the predictors (Hansen et al. 1996), which is a crucial aspect when dealing with environmental covariates. All these reasons make RF a very good tool for wind speed mapping. In each predicted location, RF produces a set of estimations, depending upon the size of the forest. All the predicted values can be used to determine the variance of the Weibull parameters, from which we can determine the local uncertainty of wind distribution.

2.4 Uncertainty Estimation

As mentioned in Sect. 2.3, RF produces a set of estimations of shape and scale for each predicted location. The higher the variance of these sets, the higher the local uncertainty. However, measuring the error of shape and scale only provides an indirect indication of the wind speed error and we are more interested in assessing the error propagation to the wind speed distribution. To assess the error propagation from the Weibull parameters to the wind speed distribution and to the mean wind-speed value, we relied on bootstrap again. By resampling with repetitions the set of values estimated by RF and computing a Weibull distribution from each resample, we were able to calculate the confidence intervals of the distribution. By repeating this process for a statistically significant number of times, we can determine the uncertainty of the wind distribution, shown in Fig. 3. In this figure, the grey histogram represents the most probable wind speed distribution, as predicted by RF, while the error bars represent the confidence interval of the estimation.

As mentioned in the introduction, wind maps usually focus on representing the mean wind speed and practitioners are used to relying on this information. Therefore, we focused on producing a measure of the mean speed error, so that the

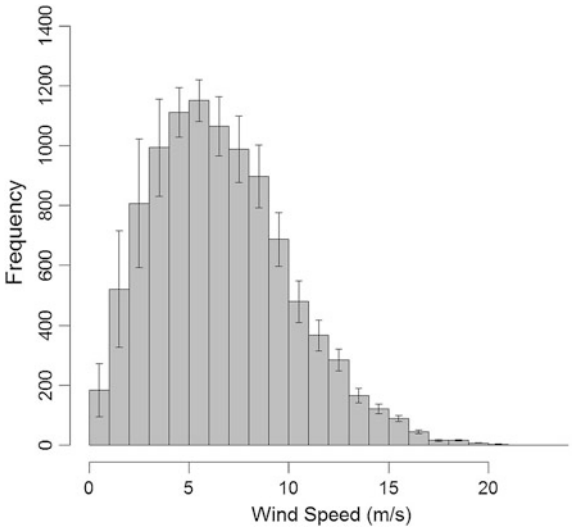


Fig. 3 Wind distribution with uncertainty estimate

map can be easily interpreted by a large majority of practitioners. We can obtain a measure of the uncertainty related to the mean wind speed by looking at the range of variation of the mean values calculated from each distribution obtained by the bootstrap. We used the Mean Absolute Deviation (AD) to compute the divergence of the wind mean speed estimates around the arithmetical average. AD is calculated according to the formula proposed by Gislason et al. (2006):

$$AD = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}| \tag{2}$$

where n is the number of wind mean speed values computed from each bootstrap set, x_i is the ith value of the wind mean speed in a set of values from 1 to n, and $\bar{x}$ is the sample arithmetical average. This parameter is more robust compared to the standard deviation, particularly for long-tailed distributions (Grimm et al. 2008). The mean absolute deviation was also used to produce the uncertainty estimate for the Weibull parameters.

2.5 Validation

Cross-validation was performed to provide a numerical validation of the estimate. We performed a 5-fold cross-validation, in which the dataset is divided into five parts (or folds). The first fold is then used for validation, while the model is fitted to the remaining four folds. This same process is then repeated until each fold is used for validation. Because the folds are chosen at random, we decided to repeat the

cross-validation process 100 times, in order to have a more reliable estimate of the test error.

The comparison between observed and predicted values was performed using the Root Mean Squared Error (RMSE), computed as follows:

$$RMSE = \sqrt{\frac{\sum_1^n \left(x_o - x_p\right)^2}{n}} \tag{3}$$

where x_o is the observed value, x_p is the predicted value, and n is the total number of samples.

3 Results and Discussion

As mentioned in the introduction, the proposed method performs the spatial prediction of wind distribution, including its local uncertainty. This is achieved by using a statistical learning approach to predict the two factors that uniquely describe the Weibull distribution. This distribution was fitted to each point of our starting dataset, shown in Fig. 1. The mapping process was performed using RF as described in Sect. 2.3. The results of the estimation are represented in the four maps shown in Fig. 4. These maps show two main areas of relatively higher values of the two parameters, located in the mountainous area of Wales and in most of Scotland. These higher values are clearly driven by elevation. However, these areas have also a relatively lower sample density and this affects the estimation accuracy of RF.

As mentioned, RF uses bootstrap resampling to create a series of trees for each location in the prediction grid. The RF estimations of the scale factor C and the shape factor k are shown in Fig. 4a, b, while the map uncertainties are depicted in Fig. 4c, d. The uncertainty is given as the mean absolute deviation of the estimates, and it is generally low, with a maximum value of 0.21 m/s, which is a third of the average test error produced by the cross-validation (Fig. 5). However, even if the deviation around the mean speed value is low it does not mean that the histogram presents a low uncertainty. For example, in the plot in Fig. 3, we used the site with the maximum AD value of 0.21 m/s and even if the deviation around the mean is very low, the uncertainty of the histogram is not negligible. This is directly propagated to the measure of the economic potential of the site and therefore needs to be properly taken into account during planning.

The validation was performed according to the approach described in Sect. 2.5; it returned a RMSE value of 0.71 m/s of the mean wind speed. This can be directly compared with the results achieved in previous studies in the UK (Cellura et al. 2008), which used several geostatistical techniques to map the mean wind speed across the same areas as in our research and used the same dataset, albeit for a different time range. They validated their methods with a leave-one-out cross-validation (Cellura et al. 2008) and concluded that the best method for estimating

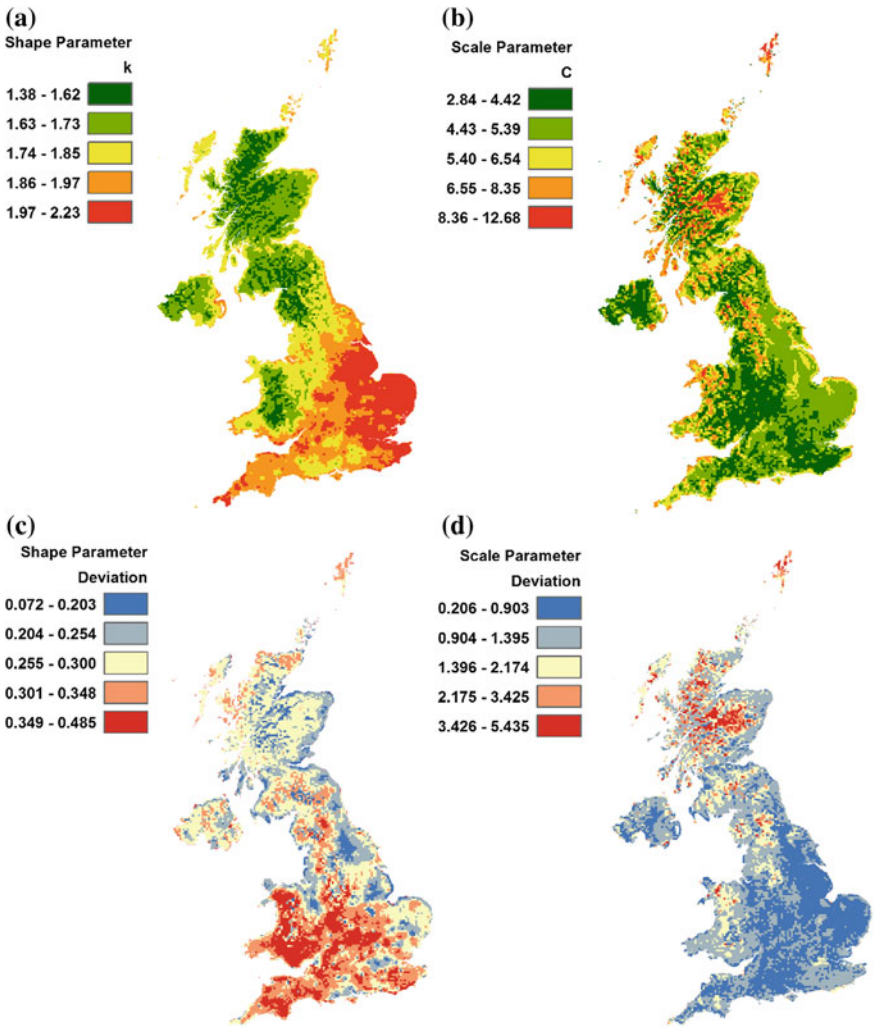


Fig. 4 **a** Spatial distribution of the shape factor *C*; **b** Spatial distribution of the scale factor *k*; **c** Uncertainty in terms of AD of the shape factor; **d** Uncertainty in terms of AD of the scale factor

the mean wind speed was co-Kriging, which returned a RMSE of 1.47 m/s. The other methods applied resulted in RMSEs above 1.61 m/s.

From published research, we also tried to compare this value with physical methods. Unfortunately, we did not find any work related to the application of these methods in the UK among the published research, we only found two that can be meaningfully compared to ours. Other articles have different ways of reporting the validation error, for example, reporting RMSE in percentages, without showing the differences in m/s), and some did not present any validation for their estimates.

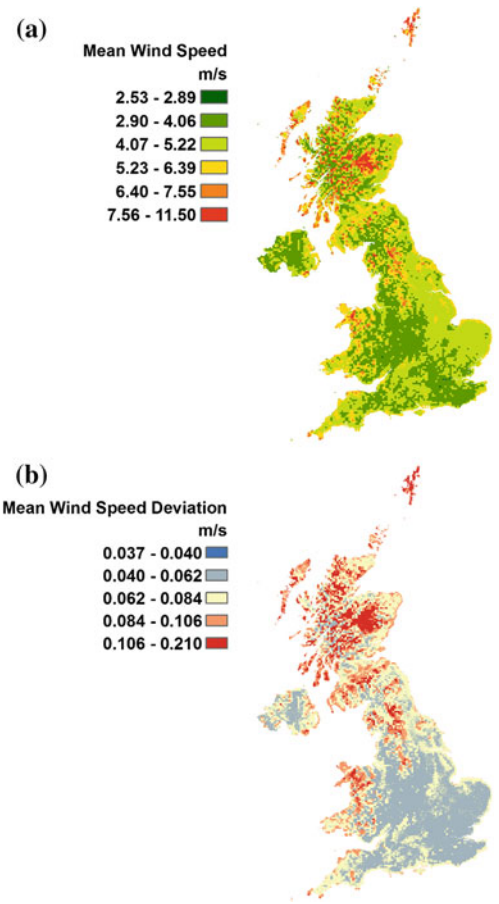


Fig. 5 **a** Map of the mean annual wind speed; **b** Map of the uncertainty estimate in terms of the AD of the wind speed

Hoaglin et al. (1983) tested the methods used in the Canadian Wind Energy Atlas, based on NWP models, starting from 10 measuring towers and estimating an area of approximately 900 km^2. They reported RMSEs of 0.74 and 0.82, depending on the resolution of the land cover data they used (higher resolution equals better results). They also did not provide a clear indication of the computational time required for the two analyses.

Gasset et al. (2012) compared all types of physical methods described in Sect. 1 on four different relatively small sites (maximum size 17 × 17 km) at a resolution of 50 m. They started from a minimum of 4 to a maximum of 9 measuring towers. They achieved the following averaged RMSE: 0.74 for CDF, 0.62 for WAsP, and 0.44 for NWP. However, the authors report that to estimate wind speed in these relatively small sites, they had to run the model for a minimum of 2.5 h (for CDF) to a maximum of 864 h (for NWP).

In this research, the training process of the statistical model and the estimation (1 km of resolution over an area of 244,119 km^2) of the same parameters generally

presented in wind speed maps took 1.81 min. The uncertainty estimation took longer, because the bootstrap needs to be run for a statistically significant number of times.

From this brief review, we can say that this method can achieve a level of accuracy comparable with physical methods, while substantially reducing the amount of time required for the analysis. This is potentially a great advantage for wind resource mapping.

It has to be pointed out that most of the tests with physical models start with very few measuring stations. The use of such a limited number of observations is pointless for statistical analysis, because the statistical learning algorithm would have too few data points for the training and this would cause a substantial reduction in its accuracy. However, we can already access large datasets with wind speed measurements sampled daily or even hourly for a large part of the world. For example, NOAA (Beaucage et al. 2014) provides a dataset with a collection of daily measurements from over 9000 stations worldwide. These data covered Europe and North America very well, so we could use the statistical learning method to produce accurate wind speed maps at high resolution, using minimum computational resources. For countries where these data are not available or where their resolution is low, we could test ways of connecting physical with statistical methods to produce high-resolution wind speed maps without the need for supercomputers.

An advantage of statistical learning methods is the possibility to compute the site-specific estimation uncertainty. The validation error we presented above, following the procedure described in Sect. 2.5, can be seen as an average value of the residuals between the measurements and the values estimated by the model. This gives a general assessment of the method's accuracy. However, for pre-construction energy estimates, this value is useless, because it is not site-specific. The local accuracy of an estimate depends on two factors, the accuracy of the model and the number of observations we have in the area. In other words, if we try to estimate wind speed using data from weather stations that are 100 km away from the planning site, chances are our estimate will have a relatively high level of error. The validation error does not give any indication about this local, site-specific error, and physical methods do not provide a way of assessing it; only statistical method can provide this information.

The map in Fig. 5b visualizes the uncertainty around the mean value for each pixel, estimated according to Sect. 2.4 and Eq. 2. In this work, the focus lies on the wind distribution and on the accurate assessment of the local accuracy of the prediction method. This aspect is crucial when the wind map is used to predict the potential for producing wind energy. For example, in Fig. 5a there are areas where the mean wind speed is high and therefore may seem attractive for exploiting the wind resource. However, these areas are also characterized by higher values of deviation of the mean wind speed, meaning that the estimates are not as reliable as in other locations. Thus, even though the calculated mean wind speed is high, building a wind farm in these areas is risky, because the accuracy of the map is low locally. More data would be needed to increase the local accuracy, and this is information that can be extracted directly from this map.

4 Conclusions and Future Work

This research introduced local uncertainties when assessing the wind speed distribution. In this work, we presented an approach based on statistical learning to spatially estimate the two parameters of the Weibull distribution. This approach allowed us to create a map of the wind speed distribution for a 5-year period. The validation results demonstrate that this method, if used properly and with the right set of covariates, can produce results comparable to physical methods in a fraction of the time. Moreover, because this method is capable of assessing its own accuracy, we were able to create a map of the local uncertainty of the estimate, which is impossible with physical methods. Furthermore, the uncertainty aspect is critical for planning wind farms, because in areas not properly covered by wind speed measurements, the site-specific accuracy of the map may well be relatively low and this means that the wind distribution may be subject to more fluctuations compared to what is reported on the map. This method allows us to pinpoint these problematic locations and warn practitioners.

This research is an example of statistical learning techniques used for wind distribution estimation. The purpose of the research is to present a method that can achieve a level of accuracy comparable to commonly applied physical approaches, while drastically reducing the computational time and, most importantly, being able to compute the uncertainty estimate. Despite the fact that our objectives were fully achieved, we are not arguing for this method to replace those based on fluid-dynamics models. However, we do think that as a research community, we should start thinking about wind estimation as a problem that would be better tackled from both perspectives. For example, we could create a workflow where few and sparsely measured data can be integrated with new estimations from complex physical methods, which would require an acceptable amount of computational time. This new dataset will then be used with the statistical method to create the final wind distribution map.

More work is clearly needed before this new workflow can be completed. We first need to explore possible methodologies to estimate wind direction distribution, which is another crucial aspect in energy infrastructure planning. Another future challenge would be to extrapolate the estimates and their uncertainty at hub height.

Approaches for this extrapolation are available from literature, but we need to test the impact of the additional steps of the existing statistical method. For example, we need to test whether it is better to extrapolate first and then estimate or if the other way around leads to more accurate predictions.

Finally, it would be interesting to calculate the propagation of the map uncertainty from the wind speed distribution to the estimate of the energy potential, to assess how the map error may influence the decision-making process when planning wind energy projects. This task will require a comparison of the long-term energy production of existing wind energy projects to the estimated energy generation using the generated maps.

Acknowledgments The authors would like to thank the UK Meteorological Office for providing the wind speed data for this research and some of the covariates. Other data providers we would like to thank are: NASA for the Aster DTM and the EU for the land-cover raster.

References

Agarwal, S. K., & Kalla, S. L. (1996). A generalized gamma distribution and its application in reliability. *Communications in Statistics. Theory and Methods, 25*, 201–210.

Akpinar, E. K., & Akpinar, S. (2005). An assessment on seasonal analysis of wind energy characteristics and wind turbine characteristics. *Energy Conversion and Management, 46*, 1848–1867.

Beaucage, P., Brower, M. C., & Tensen, J. (2014). Evaluation of four numerical wind flow models for wind resource mapping. *Wind Energy, 17*(2), 197–208.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5–32.

Cellura, M., Cirrincione, G., Marvuglia, A., et al. (2008a). Wind speed spatial estimation for energy planning in Sicily: Introduction and statistical analysis. *Renewable Energy, 33*, 1237–1250.

Cellura, M., Cirrincione, G., Marvuglia, A., et al. (2008b). Wind speed spatial estimation for energy planning in Sicily: A neural kriging application. *Renewable Energy, 33*, 1251–1266.

Center N.L.P.D.A.A. (2011). ASTER L1B. USGS/Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota.

Chan, J. C. W., & Paelinckx, D. (2008). Evaluation of random forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sensing of Environment, 112*, 2999–3011.

Climate Change Act (2008). http://www.legislation.gov.uk/ukpga/2008/27/contents

Conrad, O. (2007). SAGA—Entwurf, Funktionsumfang und Anwendung eines Systems für Automatisierte Geowissenschaftliche Analysen. Mathematisch-Naturwissenschaftlichen Fakultäten vol. Ph.D. University of Göttingen.

Cutler, D. R., Edwards, T. C, Jr, Beard, K. H., et al. (2007). Random forests for classification in ecology. *Ecology, 88*, 2783–2792.

EEA Corine Land Cover. (2006): http://www.eea.europa.eu/publications/COR0-landcover

Foresti, L., Tuia, D., Kanevski, M., & Pozdnoukhov, A. (2011). Learning wind fields with multiple kernels. *Stochastic Environmental Research and Risk Assessment, 25*(1), 51–66.

Gass, V., Strauss, F., Schmidt, J., et al. (2011). Assessing the effect of wind power uncertainty on profitability. *Renewable and Sustainable Energy Reviews, 15*, 2677–2683.

Gasset, N., Landry, M., & Gagnon, Y. (2012). A comparison of wind flow models for wind resource assessment in wind energy applications. *Energies, 5*(11), 4288–4322.

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters, 27*, 294–300.

Grassi, S., Chokani, N., & Abhari, R. (2012). Large scale technical and economic assessment of wind energy potential with a GIS tool: Case study Iowa. *Energy Policy, 45*, 58–73.

Grimm, R., Behrens, T., Märker, M., et al. (2008). Soil organic carbon concentrations and stocks on Barro Colorado Island—Digital soil mapping using random forests analysis. *Geoderma, 146*, 102–113.

Gsänger, S., & Pitteloud, J. D. (2012). World wind energy association WWEA.

Hansen, M., Dubayah, R., & Defries, R. (1996). Classification trees: an alternative to traditional land cover classifiers. *International Journal of Remote Sensing, 17*, 1075–1081.

Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (1983). *Understanding robust and exploratory data analysis* (Vol. 3). New York: Wiley.

Huang, C., Davis, L. S., & Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing, 23*(4), 725–749.

Jackson, P. S., & Hunt, J. C. R. (1975). Turbulent wind flow over a low hill. *Quarterly Journal of the Royal Meteorological Society, 101*(430), 929–955.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. New York: Springer.

Jenkins, G. J., Perry, M. C., & Prior, M. J. (2008). *The climate of the United Kingdom and recent trends*. Exeter, UK: Met Office Hadley Centre.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques.

Kwon, S. D. (2010). Uncertainty analysis of wind energy potential assessment. *Applied Energy, 87*, 856–865.

Landberg, L., Myllerup, L., Rathmann, O., et al. (2003). Wind resource estimation—An overview. *Wind Energy, 6*, 261–271.

Luo, W., Taylor, M. C., & Parker, S. R. (2008). A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from England and Wales. *International Journal of Climatology, 28*, 947–959.

Manwell, J. F., McGowan, J. G., & Rogers, A. L. (2009). Wind characteristics and resources. In *Wind energy explained—Theory, design and application* (2nd ed., pp. 43–45). New York: Wiley.

Meng, Q., Liu, Z., & Borders, B. E. (2013). Assessment of regression kriging for spatial interpolation–comparisons of seven GIS interpolation methods. *Cartography and Geographic Information Science, 40*, 28–39.

Met Office. (2012). Met office integrated data archive system (MIDAS) land and marine surface stations data (1853-current). NCAS British Atmospheric Data Centre: http://catalogue.ceda.ac.uk/uuid/220a65615218d5c9cc9e4785a3234bd0

MM5 Community Model (2015). http://www.mmm.ucar.edu/mm5/overview.html

Munteanu, I., Cutululis, N. A., Bratcu, A. I., et al. (2008). *Optimal control of wind energy systems: Towards a global approach*. New York: Springer.

Pinson, P. (2006). *Estimation of the uncertainty in wind power forecasting*. Thesis/dissertation.

Ray, M. L., Rogers, A. L., & McGowan, J. G. (2006) Analysis of wind shear models and trends in different terrain. In: *Proceedings American Wind Energy Association Windpower*.

REN21. (2012). Renewables 2012 global status report.

Rogers, A. L., Manwell, J. F., & Ellis, A. F. (2005) Wind shear over forested areas. In *Proceedings of the 43rd American Institute of Aeronautics and Astronautics Aerospace*, *Science Meeting*.

Schmidli, J., Billings, B., Chow, F. K., et al. (2010). Intercomparison of mesoscale model simulations of the daytime valley wind system. *Monthly Weather Review, 139*, 1389–1409.

Snel, H. (1998). Review of the present status of rotor aerodynamics. *Wind Energy, 1*(1), 46–69.

Susumu, S., Ohsawa, T., & Yatsu, K. (2009). A study on the ability of mesoscale model MM5 for offshore wind resource assessment in Japanese coastal waters. In *European Wind Energy Conference EWEC*.

VanLuvanee, D., et al. (2009). Comparison of WAsP, MS-Micro/3, CFD, NWP, and analytical methods for estimating site-wide wind speeds. In: Presentation from AWEA wind, 2009.

WAsP (2015). http://www.wasp.dk/

Wiesmeier, M., Barthold, F., Blank, B., et al. (2011). Digital mapping of soil organic matter stocks using random forest modeling in a semi-arid steppe ecosystem. *Plant and Soil, 340*, 7–24.

WRF Model (2015). http://www.wrf-model.org/index.php

Yim, S. H. L., Fung, J. C. H., Lau, A. K. H., et al. (2007). Developing a high-resolution wind map for a complex terrain with a coupled MM5/CALMET system. *Journal of Geophysical Research: Atmospheres*, *112*(D5), 2156–2202.

Towards a Qualitative Assessment of Changes in Geographic Vector Datasets

Karl Rehrl, Richard Brunauer and Simon Gröchenig

Abstract Changes are immanent to digital geographic vector datasets. While the majority of changes nowadays are quantitatively detectable by the use of geographic information systems their classification and impact assessment on a qualitative level with respect to specific data usage scenarios is often neglected. To close this gap, this work proposes a classification approach consisting of three parts: (1) a taxonomy for classifying quantitatively detectable edits in digital feature datasets (e.g. attribute or geometry changes), (2) a taxonomy for classifying edits into qualitative and therefore meaningful change types (e.g. feature revision or identity change) and (3) a mapping scheme providing the link from quantitative to qualitative classifications. In the context of a case study with OpenStreetMap history data the proposed classification approach is used to automatically detect and classify feature changes with respect to two feature types, namely *streets* and *buildings*, leading to a refined mapping scheme for two selected data usage scenarios, namely *vehicle routing* and *map rendering*. Results show the applicability of the approach, especially for assessing the impact of feature changes on different data usage scenarios, and provide a useful foundation for any change detection task in the context of geographic vector datasets.

Keywords Geographic vector data · Change detection · Qualitative assessment

K. Rehrl (✉) · R. Brunauer · S. Gröchenig
Salzburg Research Forschungsgesellschaft mbH,
Jakob-Haringer-Straße 5, 5020 Salzburg, Austria
e-mail: karl.rehrl@salzburgresearch.at

R. Brunauer
e-mail: richard.brunauer@salzburgresearch.at

S. Gröchenig
e-mail: simon.groechenig@salzburgresearch.at

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_11

1 Introduction

Changes are immanent to digital geographic vector datasets. Changes may occur due to missing data being added, wrong data being corrected, and real world changes being transferred to the digital dataset. Thus, before using a newer version of a previously used dataset it is useful to answer the following question: Are there any relevant or noteworthy changes between the two versions with considerable impact on an envisaged data usage scenario? Due to the variety of possible changes answering this question is not a trivial task.

For example, street network datasets are updated with periodic releases. Commonly, publishers provide the datasets without providing detailed metadata about changes. One update strategy is to substitute the whole datasets with the assumption that the quality has been sufficiently ensured by the provider. However, it can also be reasonable to gain some knowledge which parts of the dataset have been changed to which extent. Some users may be awaiting certain changes and would benefit from an easy method to examine these changes. Other users may be curious about all street segments with major revisions. For Volunteered Geographic Information (VGI) datasets this question is even more relevant due to the continuous update process by community members. If VGI datasets are used in productive applications, an in-depth analysis of changes is a crucial prerequisite prior to dataset updates in order to determine whether and in which way the update may affect an application. For example, if a navigation application uses street network data from OpenStreetMap (OSM), certain changes to the dataset (e.g. turn restrictions, speed limits, topological changes) may have significant impact on the functionality of the application.

To assess the impact of changes with respect to specific application contexts, this work introduces an approach for a qualitative classification of changes as foundation for automatic detection, interpretation and classification of arbitrary changes in digital geographic vector datasets. Since the impact of certain changes strongly depends on the envisaged data usage scenario (e.g. vehicle routing, map rendering), the proposed classification approach provides a frame for mapping quantitative edits being detected between digital geographic feature versions (e.g. type, geometry and attribute changes) to qualitative changes indicating the impact on the corresponding features. Just revealing quantitative differences (e.g. geometrical difference) of consecutive versions of digital features does not sufficiently answer the questions on the impact of the change for different application contexts. Results may be applied to any change detection task in the context of geographic vector datasets and help to separate relevant from irrelevant changes.

The reminder of this work continues with Sect. 2 on related work in the context of feature and change detection. This is followed by Sect. 3 containing definitions for changes in the real world, in representations of the real world and in implementations as well as their relationships. Section 4 introduces the two taxonomies as well as the mapping scheme and the subsequent Sect. 5 applies the approach to a case study addressing two different data usage scenarios. Section 6 concludes the work and discusses open issues.

2 Related Work

Change detection in digital datasets has been of interest to authors in many fields of research. Over the last two decades the focus has shifted from flat-file data to hierarchically structured datasets (Chawathe et al. 1996; Chawathe and Garcia-Molina 1997). In geographic data science a majority of related work is concerned with the detection of changes in remotely-sensed images (Singh 1989; Abd El-Kawy et al. 2011; Klein et al. 2012), over the last decade predominately based on object-oriented image analysis (Chen et al. 2012; Hussain et al. 2013; Blaschke et al. 2014). Concerning change detection in digital geographic vector datasets the literature review yields a considerably lower number of papers. Some work addresses changes between an older, own dataset and a newer, third party dataset for maintaining the own dataset. Von Goesseln and Sester (2005) describe change detection between different datasets as well as the integration of detected edits into own maps using automatic merging methods and an iterative closest point algorithm. Similarly, Qi et al. (2010) also detect updates of polygonal features. They compare a dataset with a more up-to-date map featuring a better resolution and consider generalization issues due to different spatial scales. Frontiera et al. (2008) apply a spatial similarity analysis during geographic information retrieval. They compare different geometric approaches including the minimum bounding box and the convex hull to assess spatial similarity. In the context of VGI, authors differentiate between attribute (tag) and geometry related updates of heavily edited OpenStreetMap features (Mooney and Corcoran 2012). Recently, Redweik and Becker (2015) proposed a concept for the detection of changes between CityGML documents. As many previous studies the approach lacks a qualitative assessment of changes.

Most of the previously mentioned approaches rely on the detection of changes in digital representations of geographic phenomena. For more than a half century, authors worked on definitions of the relationship between real world objects and their virtual counterparts, namely features based on a cognitive perspective. Gomes and Velho (1995) developed a conceptual framework with four abstraction levels. These are the physical (real-world objects), the mathematical (related to real-world objects), the representation (on a computer) and the implementation (on a computer) universe. Based on their work, Fonseca et al. (2002) introduced the five-universe paradigm consisting of the physical, the cognitive, the logical, the representation and the implementation universes. The ISO standard 19101 defines a geographic feature as "… an abstraction of a real world phenomenon" (ISO 2002). The Open Geospatial Consortium (OGC) provides with the Abstract Specification a formal conceptual model which describes the components used in geo-processing (Reed 2005). One topic of the Abstract Specification describes how features should be handled in software implementations (Kottman and Reed 2009), giving the context for the notion of digital features (digital representations of features) which is relevant to the current work. An overview of research in the field of geo-semantics is given by Janowicz et al. (2013). The authors also mention that further research

has to be investigated in the automated detection of geographically related changes being in some cases caused by environmental disasters. In this regard, qualitative aspects play a major role in the classification of feature changes. To our best knowledge a qualitative assessment of changes to geographical vector datasets has not been proposed before.

3 On the Nature of Changes in Geographic Datasets

In Geographic Information Science an observable characteristic of the real world is commonly called a *phenomenon* (Kottman and Reed 2009). A phenomenon can be an object (e.g. building, tree) or a measureable property (e.g. temperature, elevation). Geographic conceptualization describes the process of humans agreeing on *geographic concepts*, being used to represent phenomena of similar kinds (e.g. streets or buildings). Geographic concepts are typically defined in models as so called *feature types* (Kottman and Reed 2009) representing phenomena (e.g. as a map of the world) by depicting observable properties, geometries and topological relations. Model instances are called *geographic features* and may be any kind of visual or non-visual representations.

In Geographic Information Systems (GIS), features always have to be digitally represented. A *digital feature* is typically represented as a unique record in a geographic database. In order to be cognitively perceivable by humans, digital features have to be represented by different kinds of visual or non-visual feature representations (e.g. scale-dependent map renderings). Since the relationship between features and their digital representations is not a one-to-one relationship, we make a clear distinction between features and digital features. Figure 1 depicts the overall relationship between a phenomenon, a feature and a digital feature.

According to the previously outlined concepts and relationships changes may occur in any of the described *universes*: in the real world (e.g. a new street has been built), in the representation (e.g. an attribute changes the map rendering of some

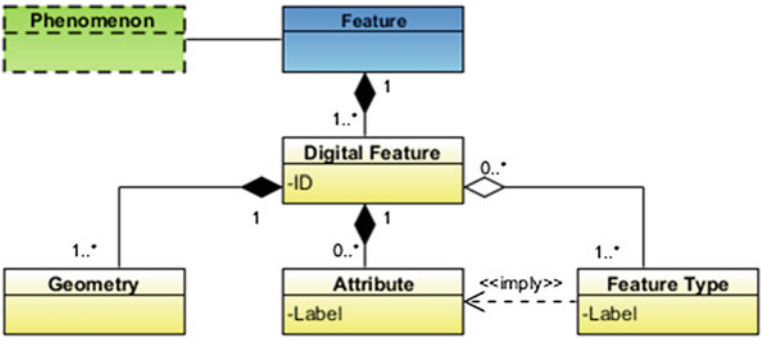


Fig. 1 The relationship between phenomenon, features and digital features

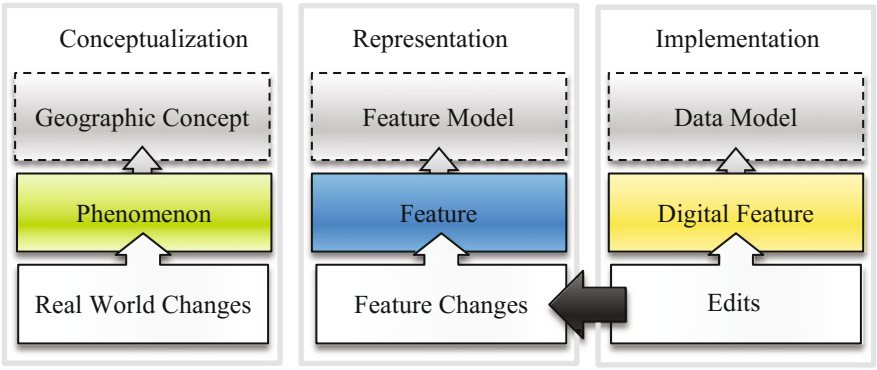


Fig. 2 The three universes of phenomena (*green*), features (*blue*), and digital features (*yellow*)

features) or in the implementation (e.g. a digital feature has been deleted). Since changes in one universe may have impact on other universes (which is one of the key issues in the later classification) we have to first provide a working definition of the relationship of the different types of changes. The proposed model (Fig. 2) is a simplification from a model originally proposed by Fonseca et al. (2002). In our working definition we condense the conceptualization process from physical objects to their cognitive representations as *Conceptualization* universe and set this universe in relation to the *Representation* and *Implementation* universes representing *Features* and *Digital Features* respectively. Consequently we define the changes as *Real World Changes*, *Feature Changes* and *Edits*. As edits we define computationally detectable changes between two versions of a digital feature. Consequently, edits may always be identified as a sequence of database operations on some database records (digital features). Figure 2 gives an overview of the relationship between the different universes and different types of changes. The grey arrow indicates that edits to digital features may have impact on representations of this feature.

To understand the impact of edits on the different universes it is crucial to introduce the concept of *feature identity*. One could argue that the identity of a feature is naturally given by the identity of the digital feature(s), the database ID. However, this kind of identity is irrelevant for identifying feature changes because this identity typically has no effect on the relationship between a feature and a phenomenon (e.g. the identity of a digital feature in a database may change independently whether the reference between the feature and the phenomenon has changed). Instead of using the digital feature identity we propose a so called *reference identity*, which is defined by the relationship between the phenomenon and its corresponding feature. For example, the relation "feature X represents phenomenon Y" gives the reference identity of feature X, and if feature X represents phenomenon Z in a later version, then the reference identity has changed but not necessarily the digital identity.

In order to clarify the proposed concept of reference identity we give two examples. A user may decide to delete a feature in a GIS because the phenomenon in the real world has been removed. Consequently, the digital feature will also be removed in the database. In this case, there is a direct relationship between a change in the real world, a change in the representation and a change in the database (although the change in the real world and the change in the representation may possess a considerable time lag). In another example, due to a database edit a digital feature gets a new digital identity. In this case the change in the digital feature universe may be automatically detected although neither the feature nor the phenomenon has changed. Since such changes are primarily detected in subsequent versions of digital features (in most cases automatically), classifying different kinds of digital feature edits to qualitative changes is the nucleus of the proposed classification outlined in the next section.

4 From Edits to Qualitative Change Types

As outlined in the previous section, changes may occur in three universes and may have different impacts on features and digital features. Changes in the real world result in out-of-date maps and changes in digital features could lead to more accurate or more detailed maps. The idea behind the proposed classification approach is not solely to rely on a quantitative change assessment (e.g. a digital feature moved 5 m to the North) but also on a qualitative assessment (e.g. a feature most likely refers to another phenomenon after an edit). For computer-based automated change detection such a classification can only be accomplished by classifying digital feature changes and giving meaning to these changes through application of heuristic rules mapping between a quantitative and a qualitative perspective. Thus, the presented classification approach consists of two steps: (1) the classification of digital feature edits (e.g. displacement of a polygon) into *quantitative edit types* (digital feature changes) and (2) the classification of edit types into *qualitative change types* (feature changes) representing the impact of edits on a feature (e.g. a revision of an existing feature or a change of the reference identity). Figure 3 shows the overall approach based on the lifecycle of digital features and the taxonomy for classifying edits (yellow) and the taxonomy for classifying edit types into change types (blue).

Digital Feature Lifecycle: In order to understand the classification approach it is worth to have a closer look at the lifecycle of a digital feature in a GIS or a geographic database (bold arrows in Fig. 3). The lifecycle of each digital feature starts with its creation. Afterwards it may run through several modifications before it is deleted again. Naturally it is also possible that a digital feature is never modified or deleted. It has to be stressed that it is not unusual that the reference identity changes during the lifecycle of a digital feature as a consequence of edits. Changing the reference identity means that a feature, although keeping the same digital identity, represents another phenomenon due to a substantial edit to the

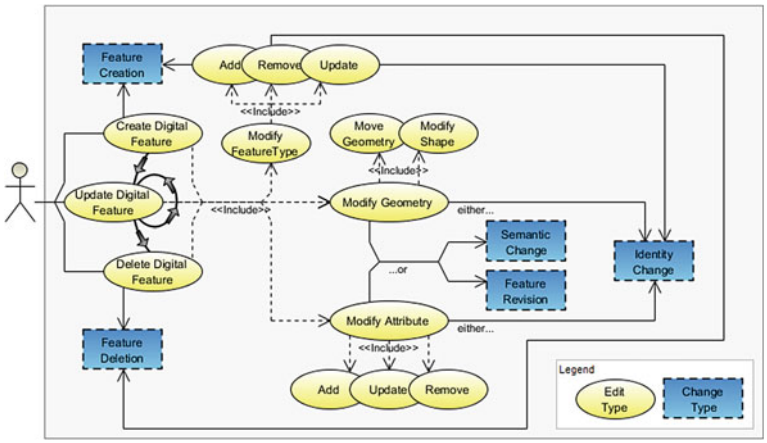


Fig. 3 Edit types (*yellow*), change types (*blue*) and their relationships; *bold arrows* indicate the lifecycle of a digital feature

geometry or the feature type of its digital counterpart. Since edits to digital features may occur at any time during their lifetime, every modification has the potential to change the reference identity. Recognizing and classifying such changes of reference identities is a crucial requirement for the following classification steps.

Classification of Edits (Step 1): The first classification step is concerned with the aggregation of atomic database operations (edits) of digital features to higher-order edit types according to the proposed taxonomy. For accomplishing this classification step it is possible to exploit the structure of the data model which is used for representing and manipulating digital features. Top level edit types such as *create*, *update*, and *delete* (Fig. 3) reveal as proper candidates for the later classification of qualitative changes. Updates may be further classified into three subclasses: *modification of feature types*, *modification of feature geometries*, and *modification of feature attributes*. Again the class *modification of feature types* and *modification of feature attributes* may be sub-classified into *add*, *update*, and *remove* type whereas the *modification of geometries* may be broken down into *move geometry* (transformation of the whole geometry) and *modify shape* (transformation of geometry parts). All sub-classes and their relationships are outlined in Fig. 3.

Classification of Changes (Step 2): Based on the edit types, the second classification step is concerned with the impact of edits on features. Firstly, the classification of the change types *feature creation* and *deletion* is rather straightforward but obviously interesting for the question whether there have been new features added to a dataset or existing features removed. Naturally, both change types have high impact on data usage for many data usage scenarios. The classification of the different modification types is more challenging, indeed: For a given data usage scenario, a change of a feature representation can be of high importance, of moderate interest, negligible or even not visible. For example, the change of a house number might be a relevant change for a navigation scenario whereas a

displacement of a building is probably not. In another usage scenario even scale may play a crucial role: displacing a feature by 10 m may be considered as significant change at a large scale while it is probably irrelevant for small scales. In this context scale is defined as a sub-property of the usage scenario and thus relevant for the impact of an edit. Reflecting on the two given examples, it gets obvious that different feature changes may have different impacts on envisaged data usage scenarios. To reflect these qualitative distinctions of changes, we propose three further change types, namely *identity change*, *semantic change*, and *feature revision* (Fig. 3). Taking creation and deletion changes into account we are now able to complete the taxonomy for classifying edits into qualitative changes with five different change types:

- **Feature Creation**: Compared to a previous dataset version, a new feature appears in the current version while the dataset has not contained another feature with the same reference identity before.
- **Feature Deletion**: Compared to a previous dataset version, a feature is missing in the current version while no new feature with the same reference identity has been created.
- **(Reference) Identity Change**: Compared to a previous dataset version, the feature reference points to another phenomenon in the current dataset version while the digital feature identity stays the same.
- **Semantic Change**: Compared to a previous dataset version, the feature still points to the same phenomenon but properties (geometry or attributes) have significantly changed in the current version and the changes have an influence on the usage of the feature with respect to a usage scenario.
- **Feature Revision**: Compared to a previous dataset version, the feature still points to the same phenomenon but properties (geometry or attributes) have changed in the current version and the changes have minor or no influence on the usage of the feature with respect to a usage scenario.

Classification Rules (Step 3): Dedicated mapping rules from quantitative to qualitative changes are the missing link for successfully accomplishing the classification task. In this regard we propose the definition of heuristic rules by considering different data usage scenarios. As foundation for deriving scenario-specific rules we propose a general mapping scheme which may be adapted accordingly. Since the mapping of creation and deletion types is rather straightforward we focus our scheme on the mapping of the specific edit types in the context of updates. Basically the scheme considers which kind of information of a digital feature (feature type, geometry, attributes) has changed in which way and which impact this change has on different feature representations. Table 1 gives an overview of the general mapping scheme.

In the following case study the proposed classification approach and the general mapping scheme are used to derive a specific rule set for classifying changes of street and building features.

Table 1 General scheme for deriving scenario-specific mapping rules from edit to change types

Edit type class	Scheme for heuristic rule →	Change type class
Modify feature type	Change to sub-type	Semantic change
	Change to other type	Identity change
Move geometry	Movement within buffer below threshold	Feature revision
	Movement within buffer above threshold	Semantic change
	Disjoint buffer	Identity change
Modify shape	Geometry type changed	Semantic change
	Change below threshold (e.g. distance, surface area)	Feature revision
	Change above threshold (e.g. distance, surface area)	Semantic change
Add attribute	Typifying attribute	Semantic change
	Specifying attribute	Feature revision
Modify attribute	Attribute similarity below threshold	Feature revision
	Attribute similarity above threshold	Semantic change
Delete attribute	Typifying attribute	Semantic change
	Specifying attribute	Feature revision

5 Case Study

In order to demonstrate the feature change type classification process, a case study has been conducted. The case study is based on parts of the OpenStreetMap (OSM) history dataset covering the city of Berlin. The main objective of the case study is to automatically detect and classify edits which occurred between July and December 2013. The used digital vector dataset has been extracted from the OSM Full History file (http://planet.openstreetmap.org/planet/experimental/) from January 4th 2014.

To show the dependency of the heuristic mapping rules on specific data usage scenarios, we investigate two distinct scenarios: *Map Rendering* and *Vehicle Routing*. Map rendering has a focus on presenting data for orientation of humans while vehicle routing has the focus to get people from A to B constrained by vehicle movement. Thus, for the case study, the two feature types *street* and *building* have been selected due to their different characteristics: Streets are typically represented as linear geometries with additional attributes for street type, name, reference id and allowed maximum speed. Buildings are typically represented as polygonal geometries with additional attributes such as building type or house number.

While the same dataset is used for both data usage scenarios, the features are processed differently. As expected, the case study indicates that geometry type, feature type, and envisaged usage have an important impact on the detailed specification of the heuristic rules and on the outcome of the change classification. Changes having a major effect on single features regarding one usage scenario may have minor or no effect regarding other usage scenarios. For example, a displacement of a building by 25 m has an effect on map rendering while it is rather irrelevant for the vehicle routing scenario. Displacing a building by 300 m however is relevant for both scenarios since not only the map rendering but also the route may change.

Methodology: The applied methodology for the case study is as follows. In a first step we extract *create*, *update* and *delete* edits from the OSM History File. To determine the edit types, all consecutive versions of all digital features are compared (Rehrl et al. 2013). The applied algorithm considers all possible changes of the digital feature, namely creates, updates, and deletes of feature types, geometries and attributes. In a second step, for the digital features of type *street* or *building* a more detailed analysis revealing the edit types is conducted. For example, a street may have been lengthened (modify shape) or a building may have been displaced (move geometry). Figure 4 shows examples of detected digital feature edits. In order to prepare the data for the qualitative classification a more detailed classification of edit types based on heuristic parameters has been defined (e.g. specifying the thresholds for geographic displacement):

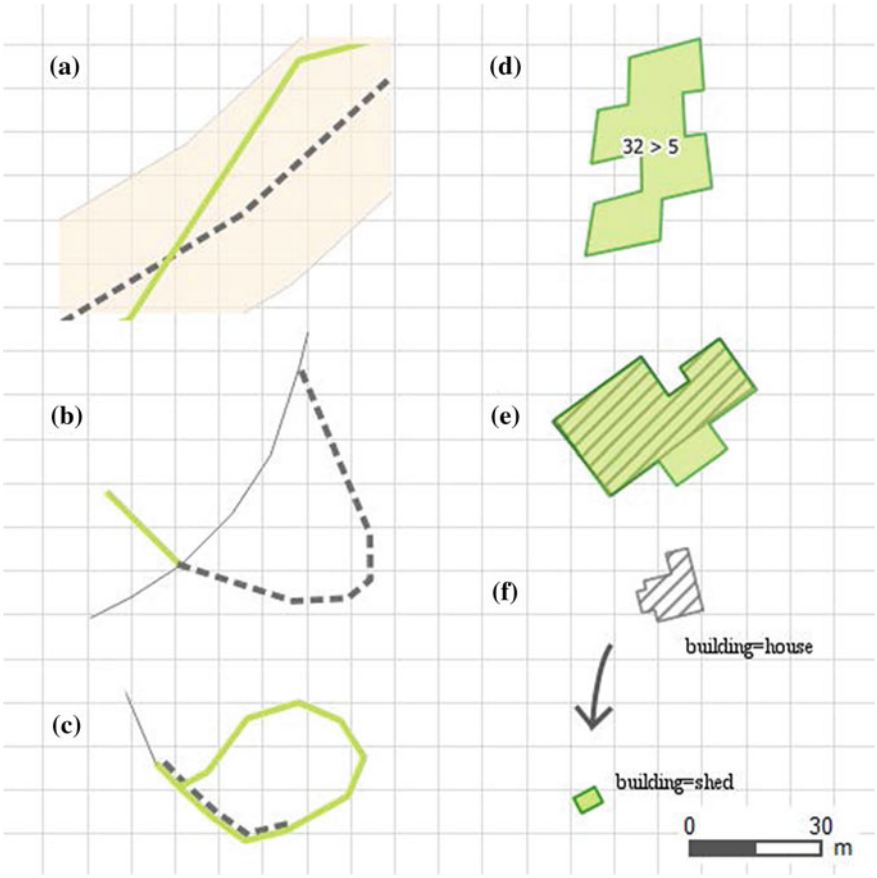


Fig. 4 Six examples of edits (*old in grey*, *new in green*): **a** Modified street within 20 m buffer, **b** modified street disjoint from old version, **c** street lengthened, **d** house number modified from 32 to 5, **e** Building shape modified, **f** building moved >30 m and the building type changed

- For comparison of streets, a buffer with a radius of 20 m around the previous geometry is used
- If the previous and the new geometry do not overlap a *Disjoint Street* edit is generated
- If a street is lengthened by more than 20 %, a *Lengthened street* edit is generated. Analogously, if a street is shortened by more than 20 %, a *Shortened Street* edit is generated
- Street type updates are categorized by *national* [motorway (link), trunk (link)], *regional* [primary (link), secondary (link), tertiary (link), unclassified] and *local* (residential, living street) categories
- For measuring the similarity of street name modifications the Levenshtein distance has been applied. If the distance is greater three, the edit is classified as *Semantic Change*, otherwise as *Feature Revision* (Zielstra et al. 2014)
- Street reference and house number updates are categorized into edits modifying the number (e.g. 7–9) and edits not modifying the number (e.g. 7–7a)
- For classifying the displacement of buildings, a buffer radius of 20 m has been defined
- Building type updates are categorized by *A* [no specified type ("yes")], *B* (apartments, hotel, house, residential, dormitory, terrace, and detached), *C* (commercial, industrial, retail, warehouse), and *D* (other types) categories.

Applying the heuristic rules on the classification of edit types leads to the results depicted in Tables 2 and 3. Table 2 lists the number of edits for the feature type *street* between July and December 2013 in Berlin. Table 3 lists the number of edits for the feature type *building* for the same timeframe and region.

Table 2 Statistical data representing edits of digital street features in Berlin between July and December 2013

Edit type	Specific edit type	Count	Filter
Modify shape	Modify within buffer	5822	
Create digital feature	Geometry created	526	
Modify shape	Street shortened	402	
Modify attribute	Type modified	239	Group change: 170
Modify shape	Street lengthened	200	
Modify shape	Modify outside buffer	192	
Delete digital feature	Geometry deleted	164	
Modify shape	Geometry split	151	
Modify attribute	Speed limit modified	73	
Modify shape	Geometry merge	68	
Modify attribute	Name modified	66	Levenshtein distance >3: 49
Move geometry	Disjoint buffers	6	More than 50 m offset: 2
Modify attribute	Reference ID modified	1	Number modified: 0

The first column contains the edit type (compare to Fig. 3) and the second the specific edit type for streets

Table 3 Statistical analysis of edits for buildings in Berlin between July and December 2013

Edit type	Specific edit type	Count	Filter
Create digital feature	Geometry created	30,148	
Modify shape	Building intersects	13,655	
Modify attribute	Type modified	1418	From A: 935; group change: 402
Delete digital feature	Geometry deleted	1359	
Modify shape	Geometry split	789	
Modify shape	Geometry merge	265	
Modify attribute	House number modified	239	Number modified: 193
Modify shape	Modify within buffer	47	
Move geometry	Modify outside buffer	6	More than 100 m offset: 2

The first column contains the edit type (compare to Fig. 3) and the second the specific edit type for buildings

The following heuristic rule set defines the mapping from edit types to change types for the two feature types *street* and *building* and both envisaged usage scenarios. The general edit types from the above taxonomies (compare with Fig. 3) are displayed in grey color, while the specific edit types tailored to the case study are in bold. The resulting change type mappings are presented italic and separated by the implication arrow.

Mapping 1 Heuristic rules for feature type *street* and usage scenario *Map Rendering*

⊤
- Create Digital Feature: **Create Street** → *Feature Creation*
- Delete Digital Feature: **Delete Street** → *Feature Deletion*
- Modify Attribute: **Type Modified**
  - Change to other group → *Semantic Change*
  - Change to same group → *Feature Revision*
- Modify Attribute: **Ref Modified**
  - Number changed → *Semantic Change*
  - else → *Feature Revision*
- Modify Attribute: **Name Modified**
  - Levenshtein distance greater 3 → *Semantic Change*
  - else → *Feature Revision*
- Modify Geometry: **Street Within Buffer**
  - Modify Geometry: **Street Shortened** → *Semantic Change*
  - else → *Feature Revision*
- Modify Geometry: **Street Crosses Buffer**
  - Modify Geometry: **Street Lengthened** → *Semantic Change*
  - else → *Semantic Change*
- Modify Geometry: **Street Disjoint** → *Identity Change*

Mapping 2 Heuristic rules for feature type *street* and usage scenario *Vehicle Routing*

⊤	
├─ Create Digital Feature: **Create Street**	→ *Feature Creation*
├─ Delete Digital Feature: **Delete Street**	→ *Feature Deletion*
├─ Modify Attribute: **Type Modified**	
│ ├─ Change to other group	→ *Semantic Change*
│ └─ Change to same group	→ *Feature Revision*
├─ Modify Attribute: **Max Speed Modified**	→ *Semantic Change*
├─ Modify Geometry: **Street Within Buffer**	
│ ├─ Modify Geometry: **Street Shortened**	→ *Semantic Change*
│ └─ else	→ *Feature Revision*
├─ Modify Geometry: **Street Crosses Buffer**	
│ ├─ Modify Geometry: **Street Lengthened**	→ *Semantic Change*
│ └─ else	→ *Semantic Change*
├─ Modify Geometry: **Street Disjoint**	→ *Identity Change*
├─ Modify Geometry: **Street Split**	→ *Semantic Change*
└─ Modify Geometry: **Street Merge**	→ *Semantic Change*

Mapping 3 Heuristic rules for feature type *building* and usage scenario *Map Rendering*

⊤	
├─ Create Digital Feature: **Create Building**	→ *Feature Creation*
├─ Delete Digital Feature: **Delete Building**	→ *Feature Deletion*
├─ Modify Attribute: **Type Modified**	
│ ├─ Change from group [A]	→ *Feature Revision*
│ ├─ Change from group [B-D]	→ *Semantic Change*
│ └─ Change within group [B-D]	→ *Feature Revision*
├─ Modify Attribute: **House number Modified**	
│ ├─ Number changed	→ *Semantic Change*
│ └─ else	→ *Feature Revision*
├─ Modify Geometry: **Building Intersects**	→ *Feature Revision*
├─ Modify Geometry: **Building Within Buffer**	→ *Feature Revision*
└─ Modify Geometry: **Building Outside Buffer**	→ *Identity Change*

Mapping 4 Heuristic rules for feature type *building* and usage scenario *Vehicle Routing*

⊤	
├─ Create Digital Feature: **Create Building**	→ *Feature Creation*
├─ Delete Digital Feature: **Delete Building**	→ *Feature Deletion*
├─ Modify Attribute: **House number Modified**	
│ ├─ Number changed	→ *Semantic Change*
│ └─ else	→ *Feature Revision*
├─ Modify Geometry: **Building Within Buffer**	→ *Feature Revision*
└─ Modify Geometry: **Building Outside Buffer**	→ *Identity Change*

Table 4 Number of classified change types per change type and usage scenario

Feature type	Usage scenario	Feature creation	Feature deletion	Identity change	Semantic change	Feature revision
Street	Map rendering	526	164	6	1017	5905
Street	Vehicle routing	526	164	6	1260	5887
Building	Map rendering	29,359	1094	6	598	14,761
Building	Vehicle routing	29,359	1094	6	193	93

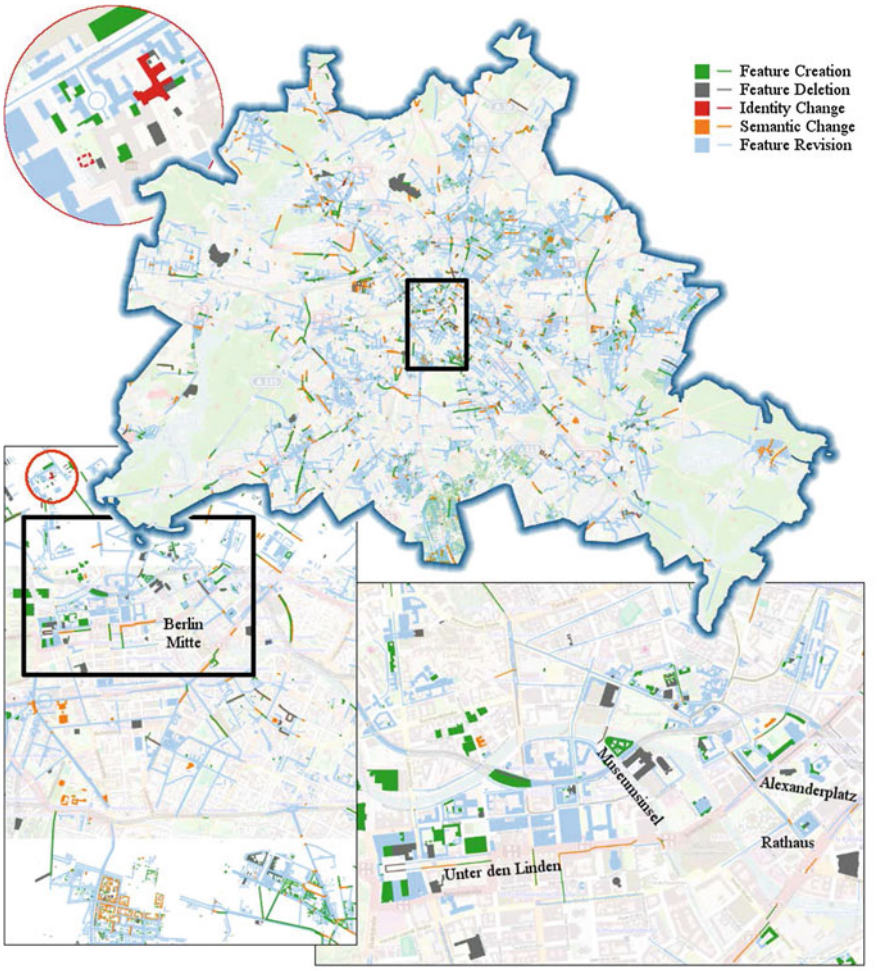


Fig. 5 Feature changes in Berlin between July until December 2013 for the usage scenario Map Rendering; styles indicate classified change types; medium- and small-scaled maps display Berlin Mitte; the circular large-scale map extract shows an *Identity Change* edit (background and map data from OSM)

Mapping the previously identified specific edit types to the qualitative change types following the above rule set results in the following change type statistics. Table 4 represents the aggregated cross tabulation for both features types and usage scenarios. For the feature type *street*, both usage scenarios reveal similar results. As expected, the most frequent change type is *Feature Revision* which occurs more than 5800 times. However, beside the high number of feature revisions, 6 *Identity Changes* have been classified. Within the same time period, more than 29,359 buildings have been created. 14,761 buildings have undergone minor revisions, which is certainly relevant for the map rendering scenario, but possibly not for the vehicle routing scenario. These results indicate that the usage scenario has a considerable impact on the classification outcome.

Figure 5 visualizes example results on differently scaled maps. The circular map extract shows a building classified as *Identity Change*. The building has been displaced in North-East direction. The distance between the buffered, old polygon and new geometry is 114 m. A cluster of *Semantic Changes* is visible on the bottom of the medium-scaled map. The type of the marked buildings has been changed from building type *residential* to the default type *building*. For data analysts using buildings of type *residential* such a change could have major impacts. *Feature Revisions* (light blue) of both *street* and *building* features are predominantly minor displacements due to position adjustments. In general, the map visualizations clearly indicate that the number of change candidates for a more detailed investigation can be reduced by the proposed qualitative classification.

6 Conclusions

Nowadays, detecting and quantifying changes in digital vector datasets is mostly feasible due to widespread storage and processing of features in geographic information systems. One of the underlying assumptions of the work is that changes can be reliably detected in digital feature versions (as atomic database operations), which has already been found in related studies. However, most previous approaches simply detect changes on a quantitative level without providing qualitative information considering the meaning of a detected change and its impact on different data usage scenarios. This work closes the gap with a qualitative classification approach and demonstrates the applicability in the context of two data usage scenarios.

Since changes may occur in three universes (phenomena, representation and implementation) it is difficult to interpret edits with the goal to derive the reason for a digital feature change. The reason could either be a change of the phenomenon (e.g. a building has been destroyed), a change of the feature representation (e.g. the map rendering has been adjusted) or simply a change in the digital feature representation (e.g. the scheme in the database has been changed). The proposed approach contributes to the assessment of changes on a qualitative level and provides hints for separating relevant from irrelevant changes with respect to a specific

data usage scenario. As we learned from the case study separating semantic and identity changes from feature revisions reduces change candidates considerably and thus a manual assessment of the remaining change candidates becomes feasible for humans.

The case study confirms the overall applicability of the approach and proposes detailed heuristic rules for the separation of relevant and irrelevant changes for the two common data usage scenarios *map rendering* and *vehicle routing*. Since the methodology for deriving heuristic rules is clearly described, the adaption to other data usage scenarios and feature types should be feasible with some extra work.

From a more critical point of view one could argue that the chosen classification rules are arbitrary which is definitely the case but also a feasible strategy for any qualitative classification approach. It has to be stressed that the proposed heuristic rules cannot be assessed on correctness, but only on adequacy with respect to a specific classification task and data usage scenario. Due to the qualitative nature, the quality assessment of classification results is subject to empirical studies with skilled participants. Conducting such studies and assessing the impact of different rulesets on classification results has been postponed to future work. There are also future plans to apply the proposed approach to other data usage scenarios, other geographic vector datasets and datasets from other regions.

Acknowledgments This work was partly funded by the Austrian Federal Ministry for Transport, Innovation and Technology.

References

Abd El-Kawy, O. R., et al. (2011). Land use and land cover change detection in the western Nile delta of Egypt using remote sensing data. *Applied Geography, 31*(2), 483–494.

Blaschke, T., et al. (2014). Geographic object-based image analysis—towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing (official publication of the International Society for Photogrammetry and Remote Sensing (ISPRS)), 87*(100), 180–191.

Chawathe, S. S., et al. (1996). Change detection in hierarchically structured information. *ACM SIGMOD Record, 25*(2), 493–504.

Chawathe, S. S., & Garcia-Molina, H. (1997). Meaningful change detection in structured data. *ACM SIGMOD Record, 26*(2), 26–37.

Chen, G., et al. (2012). Object-based change detection. *International Journal of Remote Sensing, 33*(14), 4434–4457.

Fonseca, F., et al. (2002). Semantic granularity in ontology-driven geographic information systems. *AMAI Annals of Mathematics and Artificial Intelligence, 36*(Special Issue on Spatial and Temporal Granularity), 121–151.

Frontiera, P., Larson, R., & Radke, J. (2008). A comparison of geometric approaches to assessing spatial similarity for GIR. *International Journal of Geographical Information Science, 22*(3), 337–360.

Goesseln, G., & Sester, M. (2005). Change detection and integration of topographic updates from ATKIS to geoscientific data sets. *Next generation geospatial information* (pp. 85–100).

Gomes, J., & Velho, L. (1995). Abstraction paradigms for computer graphics. *The Visual Computer, 11*(5), 227–239.

Hussain, M., et al. (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing, 80*, 91–106.

ISO. (2002). ISO 19101:2002 Geographic information—Reference model.

Janowicz, K., Scheider, S., & Adams, B. (2013). A geo-semantics flyby. *Reasoning web. Semantic technologies for intelligent data access. Lecture Notes in Computer Science* (vol 8067, pp. 230–250).

Klein, I., Gessner, U., & Kuenzer, C. (2012). Regional land cover mapping and change detection in Central Asia using MODIS time-series. *Applied Geography, 35*(1–2), 219–234.

Kottman, C., & Reed, C. (2009). The OpenGIS abstract specification, topic 5: Features.

Mooney, P., & Corcoran, P. (2012). Characteristics of heavily edited objects in OpenStreetMap. *Future Internet, 4*(1), 285–305.

Qi, H. B., et al. (2010). Automated change detection for updating settlements at smaller-scale maps from updated larger-scale maps. *Journal of Spatial Science, 55*(1), 127–140.

Redweik, R., & Becker, T. (2015). Change detection in CityGML documents. In *3D Geoinformation science*. *Lecture Notes in Geoinformation and Cartography 2015* (pp. 107–121). Springer International Publishing. .

Reed, C. (2005). The OpenGIS abstract specifications, Topic 0—Overview.

Rehrl, K., & et al. (2013). A conceptual model for analyzing contribution patterns in the context of VGI. In J. Krisp (Ed.), *Progress in location-based services. Lecture Notes in Geoinformation and Cartography* (pp. 373–388). Springer, Berlin.

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing, 10*(6), 989–1003.

Zielstra, D., et al. (2014). Areal delineation of home regions from contribution and editing patterns in OpenStreetMap. *ISPRS International Journal of Geo-Information, 3*(4), 1211–1233.

Part III
Understanding and Improving Mobility

Usage Differences Between Bikes and E-Bikes

Dominik Allemann and Martin Raubal

Abstract A high share of bicycle traffic in urban areas can be advantageous in order to tackle traffic related problems such as congestion, over-crowded public transportation or air pollution. Through an increased dissemination of e-bikes in recent years, cycling has become a viable transportation alternative for an even broader audience. The consequences of this trend on urban mobility are not yet clear. In order to get a clearer picture, one first needs to understand the major usage differences between e-bikers and cyclists. In this paper we demonstrate how a first insight into these differences can be gained by analysing GPS tracking data, recorded within the context of a field study. E-bikers as well as conventional cyclists prefer riding on any kind of bike trail whilst e-bikers rather choose bike trail types with a larger exposure to vehicular traffic. Taking a minimal distance route was the most important route choice factor for both cyclists and e-bikers. E-bikers perceived their rides to be slightly more safe and convenient as compared to conventional cyclists.

Keywords Urban mobility · Route choice · Tracking data · Bikes · E-bikes · Field study

1 Introduction

In recent years the on-going spread of urban areas (United Nations 2012), the amplification of traffic related inconveniences such as congestions and air pollution, the increasing consumption of cultivated land as well as intense discussions about

D. Allemann (✉) · M. Raubal
Institute of Cartography and Geoinformation, ETH Zurich, 8093 Zurich, Switzerland
e-mail: allemanndo@gmail.com

M. Raubal
e-mail: mraubal@ethz.ch

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_12

energy security can be observed. In this context, approaches for a more sustainable way of mobility are receiving more and more recognition.

An approach to tackle the above-mentioned challenges is the promotion of a higher bicycle mode share. Cycling as an alternative for utilitarian trips may help to reduce congestions and the emission of air pollutants as well as greenhouse gases. It furthermore contributes to public health (De Hartog et al. 2010; Dill 2009). According to Heinen et al. (2010, 2013) the determinants to use bicycles for utilitarian trips are manifold. For all advantages of cycling as a mode of urban transportation, it is only appropriate for moderately sporty people and flat environments. Using a bike for utilitarian purposes is unattractive in hilly areas and for people with limited physical endurance. With pedal-assisted e-bikes cycling becomes available to a broader audience and an alternative for urban transportation even in hilly areas (DeMaio 2009; Parkes et al. 2013; Shaheen et al. 2010).

Despite the increasing dissemination of e-bikes in the last few years there has not been a study until now which investigates how cyclists and e-bikers differ concerning their route choice behaviour and the perception of the chosen routes. The understanding of usage differences between bikes and e-bikes may result in the adaption of existing bike trail networks and influence the planning of new bike trails as well as urban transportation policies (Rose 2012). It can also be useful for the design of two-wheeler sharing systems with e-bikes incorporated (Cherry et al. 2011). In this paper, the study of usage differences between bikes and e-bikes is further broken down into four research goals:

1. Analysis of general trip characteristics such as length and mean velocity for bikes and e-bikes.
2. Analysis of differences regarding the preferences for different types of bike trails.
3. Analysis of differences regarding the factors influencing a cyclist's and an e-biker's route choice.
4. Analysis of differences regarding the perception of a ride in terms of physical activity, convenience and safety.

The remainder of this paper is structured as follows. After the discussion of the relevant literature in Sect. 2, the chosen approach to evaluate differences between bike and e-bike usage based on GPS tracking data and rider's interviews is explained in Sect. 3. Section 4 deals with the discussion of the results. Section 5 presents conclusions and directions for future work.

2 Related Work

Travel behavior research deals with a variety of questions ranging from aspects about wayfinding and the perception of geographic space to the destination selection and the choice of modes of transportation (Golledge and Stimson 1997; Golledge 1995; Montello 2005).

Travel and mobility behavior can be assessed based on travel diaries and questionnaires or tracking data (Van Evert et al. 2006; Wolf 2006). The use of GPS tracking instead of travel diaries in the context of travel behavior research leads to a more accurate knowledge about travel times, distances as well as geolocation of origin and destination of a trip (Stopher et al. 2002; Bohte et al. 2008). A broad range of studies uses GPS data to assess movement patterns and travel behavior. For example, Van der Spek (2008) and Van der Spek et al. (2009) use GPS tracking data for the analysis of pedestrian movements. Chung and Shalaby (2005) as well as Tsui and Shalaby (2006) studied GPS tracks with respect to transportation modes used and developed a GIS-based tool. Cellina et al. (2013) make use of a smartphone-based GPS tracking approach to understand barriers towards the adoption of electric vehicles and to foster a change in thinking towards a more sustainable mobility.

Several studies analyze relevant route choice factors for cyclists by developing formal route choice models based on Stated Preference (SP) Surveys. Brick et al. (2012) reported travel time and type of infrastructure to be the most critical factors in determining route choice. The most preferred type of cycling infrastructure were facilities segregated from traffic. Routes without any kind of cycling infrastructure were least preferred. Hunt and Abraham (2007) reported similar findings: cycling travel times along segments with mixed traffic are perceived more onerously compared to cycling travel times along bike lanes and bike paths. Stinson and Bhat (2003) analyzed eleven route choice factors. Low travel times, a low traffic volume and a high degree of separation from vehicular traffic were found to belong to the three most important route choice factors for cyclists. Furthermore, the analysis showed that cyclists prefer having few traffic lights and stop signs along the chosen route. Fajans and Curry (2001) also discussed this aspect from an energetic point of view. Hunt and Abraham (2007), Stinson and Bhat (2005) as well as Taylor and Mahmassani (1996) reported the dependency of the importance of route choice factors according to a cyclist's experience. Experienced cyclists were found to be more sensitive to factors such as travel time and less sensitive to factors relating to the safety of the trip as compared to inexperienced cyclists. Inexperienced cyclists however, preferred routes with a continuous cycling facility and more traffic lights. For experienced as well as inexperienced cyclists the separation from motorized traffic ranked among the most important route choice attributes. By analyzing the perception of cycling depending on the level of experience, Rondinella et al. (2012) revealed that the more the bicycle was used, the more positive the cycling experience was.

Only a few studies have analyzed the route choice behavior of cyclists based on effectively chosen routes. Aultmann et al. (1997) asked study participants to trace routes travelled by bicycle on a map. After digitization of these routes, they were compared with corresponding shortest distance routes using a Geographic Information System (GIS). Shortest path routes incorporated more street segments with a slope and more turns per kilometer as compared to the chosen routes. Harvey and Krizek (2007) investigated the commuter bicyclist behavior based on GPS tracking data and an interview of cyclists. They found the perceived safety to have a

significant influence on the trip speed. As perceived safety decreases the cyclists also ride more slowly. As the confidence in a route decreases also the trip speed decreases. Broach et al. (2012) and Menghini et al. (2010) developed bicycle route choice models based on GPS tracking. Broach et al. (2012) reported that the attributes distance, turn frequency, slope, intersections, facility types and traffic volumes contribute to a route's attractiveness to bicyclists. Menghini et al. (2010) emphasized that cyclists try to avoid intersections with traffic lights and route segments with a steep maximum gradient. Furthermore, cyclists appreciate direct and marked routes.

Research about e-bikes is until now quite limited. Dill and Rose (2012) interviewed 28 e-bike riders in Portland, Oregon, US about the usage of their e-bikes. A majority of the participants stated that the e-bike had been purchased as an alternative to a car or to overcome their limited abilities to ride a conventional bike. They further reported an increased overall amount of cycling since the e-bike purchase. Despite the electric propulsion, the level of physical activity associated with an e-bike ride was found to be satisfactory by a vast number of e-bike owners. Many of the interviewed e-bike users felt safer and more stable while riding with an e-bike than on a conventional bike. Accompanied with this perception, one could interpret the fact that one third of the interviewed e-bike users found their vehicle enabled them to operate as if they were using a motor vehicle. In a similar survey, Popovich et al. (2014) interviewed 27 e-bike users in the Sacramento, California area. The mentioned reasons for purchasing an e-bike and positive aspects of riding an e-bike mostly correlate with those reported by Dill and Rose (2012). However, participants of the study by Popovich et al. (2014) raised concerns about the risk of theft and doubts of using paths together with bikers and pedestrians. Also the weight of e-bikes was mentioned as a negative point.

No studies exist investigating exclusively usage differences between bikes and e-bikes. Weinert et al. (2008) surveyed bike and e-bike users in the Chinese city of Shijazhuang. However, they tried to analyze why more and more bike and public transit users have shifted their mode of transportation to e-bikes. As reasons for shifting from bike to e-bike, the surveyed persons stated a higher convenience, flexibility and comfort while riding an e-bike.

3 Evaluating Differences Between Bikers and E-Bikers

3.1 Data Collection

Differences between bikers and e-bikers were investigated on the basis of GPS tracking data, collected in the context of a field study conducted in the city of Zürich, Switzerland. Participation in the field study was theoretically possible for everybody. However, the recruitment of participants focused on employees and students of ETH Zurich. Emails, advertising the field study, were sent to around 440 people.

Study participants were asked to ride with either a bike or an e-bike from the ETH campus Hönggerberg to the ETH campus Zentrum and back while being tracked with a GPS tracking device. The distance between the two campuses is approximately 6 km and there is an elevation difference of about 60 m with steep parts as well as relatively flat parts. The location of the two campuses in Zürich is shown in Fig. 1.

The used e-bike assisted the rider's pedal power up to 25 km/h. According to Swiss traffic regulations it is treated as a conventional bike. Study participants were not asked to follow a specific route. Among the group of e-bikers, 36.6 % of the participants voluntarily informed themselves about possible routes prior to the ride and 63.3 % have ever ridden a two-wheeler between the two campuses. Among the group of bikers, 60 % of participants voluntarily informed themselves about possible routes prior to the ride and 50 % have ever ridden a two-wheeler between the two campuses.

After a test ride the field study participants had to fill in a questionnaire that was structured into three parts. In the first part, participants were asked to provide some basic information such as age, zip code of their domicile, and their familiarity with biking and e-biking. The second part referred to the ride from ETH Hönggerberg to ETH Zentrum (further referred to as "downhill ride"). On the one hand, participants were asked to rate the following two statements regarding the perception of the chosen route on a four-level Likert scale ranging from being highly true to being not true at all:

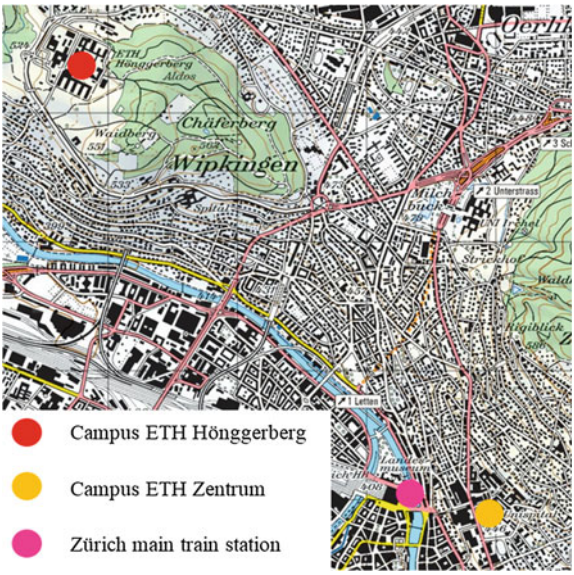


Fig. 1 Location of ETH Hönggerberg and ETH Zentrum in the context of Zürich. Background pixel map: © 2013 swisstopo (JD100042)

1. The chosen route was perceived to be convenient and safe.
2. The chosen route was perceived to be associated with a high level of physical activity.

On the other hand, participants were asked to rate the importance of eight different factors for their downhill ride route choice, also on a four-level Likert scale ranging from very important to not important at all. The eight factors are listed in Table 1.

The third part of the questionnaire was structured in the same way as the second part. However, it referred to the ride from ETH Zentrum to ETH Hönggerberg (further referred to as "uphill ride").

In total 21 persons participated in the field study. Ten of the participants rode a bike, 11 rode an e-bike. Table 2 summarizes descriptive statistics of the field study participants.

Table 1 Route choice factors presented to field study participants

Route is known from previous rides with a two-wheeler
Route is known from public transit trips
Route has a minimal distance
Route has a high share of cycling lanes and paths
Route has few traffic lights
Route avoids extraordinarily steep segments
Route has a low traffic volume
Route is attractive (in terms of landscape and view)

Table 2 Descriptive statistics of field study participants

	Group of bikers (10)	Group of e-bikers (11)
Mean age	23.1 years	28.1 years
Median age	22 years	25 years
Living	40 % in Zürich	54.55 % in Zürich
	60 % out of town	45.45 % out of town
Ever ridden a two-wheeler between ETH Hönggerberg and ETH Zentrum	50 % yes	63.6 % yes
	50 % no	36.4 % no
Familiarity with bikes (Min. = 1/Max. = 4)	4	3.73
Familiarity with e-bikes (Min. = 1/Max. = 4)	1.7	1.64

Table 3 Bike trail categorization

Category ID	Description
1	Official route without further measures; only visible through signposts
2	Official route along streets with low traffic
3	Physically separated bike lane
4	Visually separated bike lane
5	Official route along streets with a traffic ban

3.2 Enrichment of Street Network Data

Street network data from the Federal Office of Topography of Switzerland *swisstopo*[1] was used to evaluate the GPS tracks. The network dataset was manually enriched with information about official bike routes and the kinds of bike trails available. Five different types of bike trails were differentiated. Table 3 provides a description of the five categories.

By reviewing the raw tracking data it was determined for which street network segments information about bike trail availability was necessary. For the network segments in question the availability of one of the five mentioned bike trail categories was determined using *Google Street View*[2] and Zürich's online city map.[3]

The network dataset was also enriched manually with information about the location of traffic lights. Like the information about the bike trail availability, the positions of traffic lights were only collected for the relevant parts of the network. This network enrichment task was based on *Google Street View* and the authors' awareness of the local conditions.

3.3 Map-Matching of Tracking Data

The GPS tracks were map-matched to the street network using a geometrical, post-processing approach as described by Dalumpines and Scott (2011). The approach involves geometric and topological GIS functionalities. A buffer is created around a raw GPS trajectory (purple polygon around highlighted red line in Fig. 2). Inside this buffer the shortest path on the street network is tried to be found. The start- and endpoint of the trajectory are snapped to the next network element defining the start- and endpoint for the shortest path search. Afterwards, the buffer boundary is intersected with the street network segments. These intersection points are stored as barriers. The barriers are used for the shortest path search in order to prevent that

[1] http://www.swisstopo.ch/.

[2] http://www.maps.google.ch/.

[3] http://www.stadtplan.stadt-zuerich.ch/.

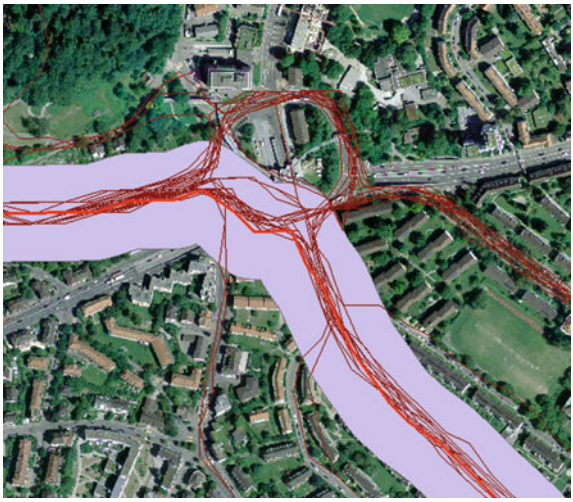


Fig. 2 Buffer polygon (*purple area*) around unmatched GPS track (*highlighted in red*). Background orthophoto: © 2013 swisstopo (JD100042)

street network elements outside the buffer region are considered being part of a shortest path (Fig. 3). If a shortest path could be found it is appended to a feature class containing all map-matched tracks. The map-matched tracks contain information about the share of the individual bike trail categories on the total route length as well as the number of traffic lights along the route. Figure 3 shows the example of an unmatched track (red line) overlaid by its corresponding matched track (blue line).

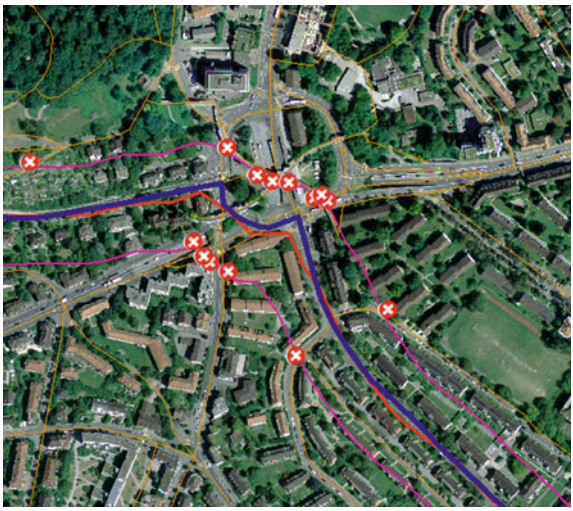


Fig. 3 Unmatched GPS track (*red line*) overlaid by matched track (*blue line*) and generated barriers (*white crosses*). Background orthophoto and street network: © 2013 swisstopo (JD100042)

4 Results

4.1 General Trip Characteristics

4.1.1 Velocity

The average bike and e-bike trip velocity of the downhill trips did not differ significantly. While riding downhill the average velocity of e-bikers was 23.7 km/h and the average velocity of bikers was 24 km/h. Especially on uphill rides, the electric propulsion of the e-bikes is reflected in the mean trip velocities. At a 20 % level of significance ($p = 0.15$) e-bikers were found to ride significantly faster (average trip velocity 19.6 km/h) than conventional bikers (average trip velocity 17.2 km/h) while riding uphill.

4.1.2 Length

The downhill trip lengths as well as the uphill trip lengths of bikes and e-bikes did not differ significantly from each other. However, the analysis of the tracking data showed that from a general point-of-view the downhill trip lengths spread over a wider range than the uphill trip lengths. The minimal and maximal trip lengths for downhill and uphill trips of both, bike and e-bike rides, are shown in Table 4.

For all field-study participants the downhill ride took place before the uphill ride. Therefore, it can be assumed that spatial learning effects led to the lower spread of trip lengths for the uphill rides as compared to the downhill rides.

4.1.3 Traffic Lights

The number of traffic lights occurring along a certain route was set into relation to the corresponding trip length. Neither for the downhill rides nor for the uphill rides significant differences regarding the occurrence of traffic lights between bikes and e-bikes were found. Comparing all e-bike trips (downhill as well as uphill) with all bike trips, the data showed that traffic lights occur at a 20 %-level of significance ($p = 0.16$) more often on e-bike tracks than on bike tracks.

Table 4 Summary of field study trip lengths

	Bikes		E-bikes	
	↓ Downhill (km)	↑ Uphill (km)	↓ Downhill (km)	↑ Uphill (km)
Maximum route length	8.131	6.120	6.129	5.901
Mean route length	5.985	5.823	5.686	5.831
Minimal route length	5.472	5.472	5.472	5.687
Standard deviation	0.800	0.200	0.243	0.066

4.2 Bike Trail Preferences

Figure 4 shows the percentage distribution of the recorded downhill kilometers among the investigated types of bike trails. The category "total on bike trails (w/o category 1)" ignores kilometers recorded along streets where bike trail category 1 (no measures but official route) occurs only on one side.

The number of kilometers e-bikers rode on official routes without any measures was found to be significantly higher at a 10 % level of significance ($p = 0.08$) than the corresponding number of bike kilometers. The category "no measures but official route" mostly occurs along streets where a higher traffic volume can be found than, for example, on routes leading through streets with low traffic.

The preferences of the e-bikers and cyclists for the different bike trail categories for the uphill rides are comparable with those of the downhill rides, as can be seen in Fig. 5. E-bikers showed again a higher preference for the categories "no measures but official route" and "visually separated lane" whilst cyclists tended to prefer routes along streets with a traffic ban.

For the categories "no measures but official route" and "route along traffic ban", the preference differences between bikers and e-bikers were found to be significantly different at a 20 % level of significance ($p = 0.16$ and $p = 0.19$). For all other categories the differences between bikes and e-bikes were not found to be statistically significant.

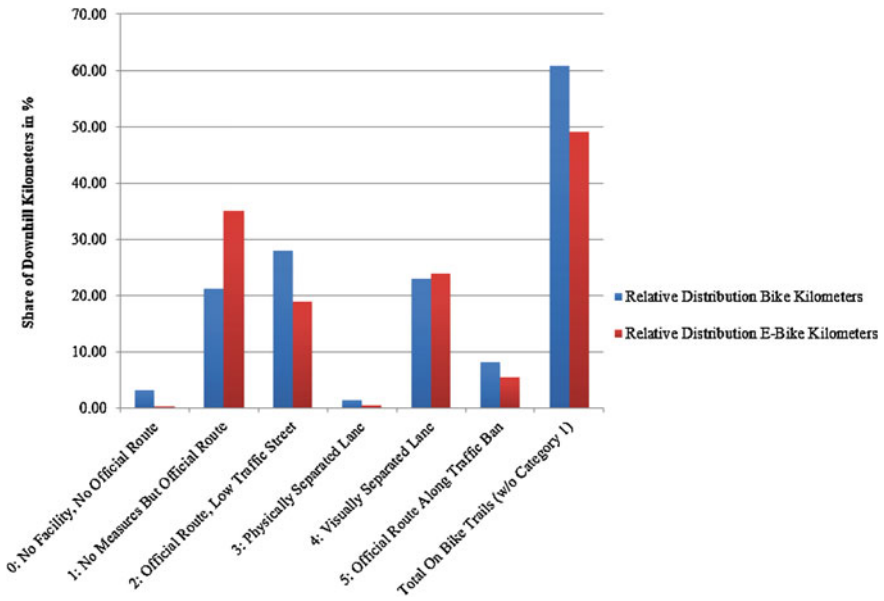


Fig. 4 Distribution of observed downhill kilometers among bike trail categories

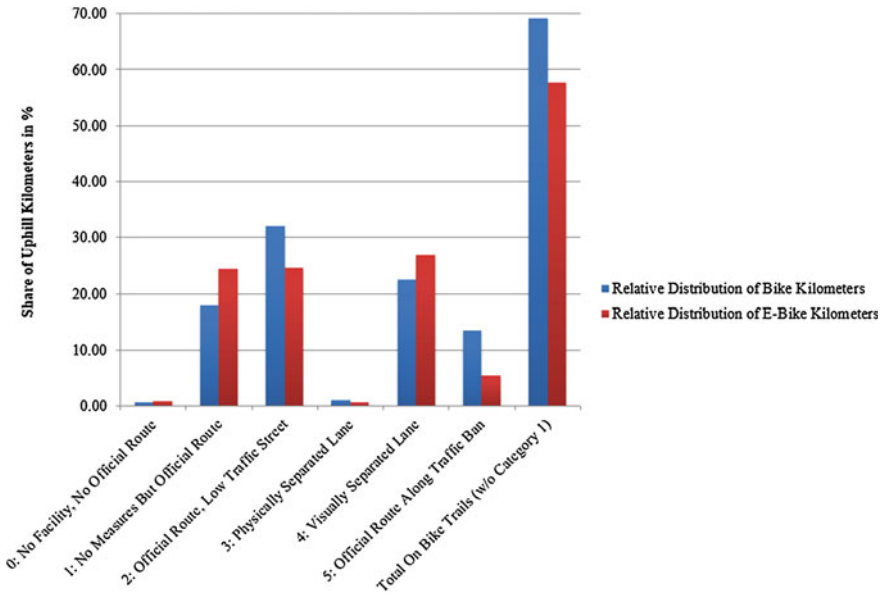


Fig. 5 Distribution of observed uphill kilometers among bike trail categories

Considering the general differences between bikes and e-bikes, disregarding whether a track was recorded while riding uphill or downhill, many of the already stated differences and assumed preferences between bikes and e-bikes were found to be statistically significant. Compared to cyclists, e-bikers covered on a 10 % level of significance more kilometers on official routes, which are not equipped with any further measures for cycling ($p = 0.04$) and along cycling lanes which are not physically separated from vehicular traffic ($p = 0.06$). Also at a 10 % level of significance ($p = 0.09$) cyclists showed a higher preference for riding on cycling routes with a ban for vehicular traffic. Furthermore, cyclists chose significantly more often (20 % level of significance) to ride along physically separated cycling lanes ($p = 0.18$), along streets with low traffic ($p = 0.16$) and generally along routes with a high share of any kind of cycling facilities ($p = 0.16$) as compared to e-bikers.

4.3 Route Choice Factors

The field study participants were asked to rate eight route choice factors on a Likert scale between 1 (= not important) and 4 (= important). Table 5 shows the ranking of the relative importance of each factor for cyclists and e-bikers but also the averaged absolute stated importance for the downhill rides.

Table 5 Relative and absolute ranking of route choice factors for downhill rides

Rank	Bikes	Absolute	E-bikes	Absolute
1	Minimal distance	3.1	Minimal distance	2.73
2	High share of bike lanes and paths	2.7	Avoidance of steep road segments	2.64
3	Low traffic volume	2.7	Low traffic volume	2.55
4	Known from previous rides with a two-wheeler	2.4	Known from public transport	2.45
5	Known from public transport	2.2	Less traffic lights	2.3
6	Nice in terms of landscape, view etc.	2.1	Known from previous rides with a two-wheeler	2.27
7	Less traffic lights	1.9	High share of bike lanes and paths	2.2
8	Avoidance of steep road segments	1.9	Nice in terms of landscape, view etc.	1.91

Among all route choice factors, the preference difference between cyclists and e-bikers was only significant for the factor "avoidance of extraordinarily steep road segments" ($p = 0.14$). Even though the terminology "extraordinarily steep road segment" was not defined, one can state that the avoidance of steep route segments is more important for e-bikers than for bikers. A final statement about the reasons leading to this preference is neither possible with the available data nor can this observation be generalized. One interpretation is that the higher weight of the e-bike in comparison to the conventional bike leads to a larger need to have control over the e-bike on steep route segments. Since the circumnavigation of steep route parts with an e-bike is associated with less physical effort than with a conventional bike, e-bikers place more value on the avoidance of steep areas.

Minimizing the distance seems to be the most important factor for both bikers and e-bikers from a relative point-of-view while riding downhill. From an absolute point-of-view taking a route with a minimal distance is for e-bikers slightly less important than for bikers. Deviating from the shortest path means less physical effort for an e-bike rider than for a conventional cyclist due to the aid of the electric propulsion. The occurrence of "low traffic volume" among the three most important factors for bikers and e-bikers can be interpreted that both are well aware of belonging to the weakest traffic participants. Whilst a high share of bike lanes along the chosen route is the second most important route choice factor for cyclists, this factor is ranked at second-to-last place among the group of e-bikers. The finding of Dill and Rose (2012) that e-bikes enable their riders to operate more like motor vehicles and its riders also feel safer and more stable than bikers, might explain the comparable low ranking of the factor "high share of bike lanes". Both bikers and e-bikers gave a relatively low priority for a route being attractive in terms of landscape and view. This indicates that the participants perceived the test rides rather as a commuting than a recreational trip.

Table 6 Relative and absolute ranking of route choice factors for uphill rides

Rank	Bikes	Absolute	E-bikes	Absolute
1	Minimal distance	3.6	Minimal distance	3
2	High share of bike lanes and paths	2.9	Known from previous rides with a two-wheeler	2.36
3	Low traffic volume	2.8	Known from public transport	2.36
4	Known from previous rides with a two-wheeler	2.7	Low traffic volume	2.36
5	Avoidance of steep road segments	2.6	High share of bike lanes and paths	2.27
6	Known from public transport	2.2	Less traffic lights	2.27
7	Nice in terms of landscape, view etc.	2.1	Nice in terms of landscape, view etc.	2.18
8	Less traffic lights	2	Avoidance of steep road segments	2

The ranking of the route choice factors for the uphill rides is shown in Table 6. Cyclists rated the same three factors as for the downhill rides to be the most important ones. Furthermore, as for the downhill rides, taking a route with minimal length seems to have the highest importance for all participants, no matter whether e-biker or cyclist. However, from an absolute point-of-view e-bikers pay (at a 20 %-level of significance; $p = 0.17$) less priority to taking the shortest route than cyclists do. Furthermore, the group of cyclists put a significantly higher priority to having a high number of bike lanes and bike paths while riding uphill (20 % level of significance; $p = 0.11$).

The "avoidance of extraordinarily steep segments" is the least important factor for e-bikers. At the same time the "avoidance of extraordinarily steep segments" is for bikers more important for the uphill ride than for the downhill ride. Notably, for e-bikers it is exactly the other way around. These facts support the assumption that the electric propulsion makes an e-bike ride more convenient than a bike ride.

4.4 Route Perception

The field study participants were asked to rate how they perceived the downhill as well as the uphill ride in terms of physical activity, safety and convenience. Table 7 gives an overview of how the participants on average rated their rides with respect to the perceived safety and convenience.

Regarding the perceived safety and convenience of the chosen route, no statistically significant differences were found between cyclists and e-bikers regardless of trip direction. The cyclists perceived the downhill ride slightly less convenient and safe as compared to the e-bikers. Among the group of e-bikers the uphill trip was perceived safer and more convenient compared to the downhill trip.

Table 7 Average perception of safety and convenience (1 = not safe and convenient at all; 4 = extremely safe and convenient)

	Bikes		E-bikes	
	↓ Downhill	↑ Uphill	↓ Downhill	↑ Uphill
Safety and convenience	2.6	2.6	2.64	2.82

Table 8 Average perception of physical effort (1 = low physical effort; 4 = high physical effort)

	Bikes		E-bikes	
	↓ Downhill	↑ Uphill	↓ Downhill	↑ Uphill
Physical activity	1.8	2.7	1.55	1.82

The slight differences in the perception of safety and convenience between bikers and e-bikers match findings reported by Dill and Rose (2012). Furthermore the support by the electric propulsion leads to added convenience while riding an e-bike. However, these differences contradict findings reported by Stinson and Bhat (2005) that less-experienced riders perceive routes to be less safe than more experienced riders do. Compared to the group of bikers, the group of e-bikers stated marginally lower familiarity with both e-bikes and bikes. Because the differences are not statistically significant, a concluding evaluation is not possible.

In terms of physical activity, the cyclists perceived the downhill trips as being slightly more strenuous than the e-bikers did. Riding uphill, the e-bikers rated the physical effort at a 5 % level of significance ($p = 0.03$) lower than bikers. Table 8 shows the numerical background of the mentioned aspects.

From a general point-of-view, e-bike trips were perceived as being less strenuous than bike trips. The differences in the perception of the physical effort between e-bikers and bikers are smaller for downhill rides than for uphill rides. The findings of the perception of the physical effort fit to the already-mentioned assumption of added convenience of e-bike rides in comparison to conventional bikes due to the electric propulsion. Especially while riding uphill, the electric propulsion has a definite advantage.

5 Conclusions and Directions for Future Work

The analysis of the tracking dataset pointed to a tendency that e-bikers ride more often along routes with a higher exposure to vehicular traffic. The rider's interviews supported this tendency: E-bikers rated "low traffic volume" to be a less important route choice factor than cyclists. The theory of the lower aversion of e-bikers to vehicular traffic is furthermore supported by the fact that more traffic lights occur per ridden kilometer along the routes chosen by e-bikers compared to the routes

chosen by cyclists. In the urban context, traffic lights most often occur on streets with a higher volume of vehicular traffic.

Bikers as well as e-bikers rated "minimal distance" to be the most important route choice factor. Despite the electric propulsion of the e-bike, rides with it are obviously still related with physical activity. However, especially while riding uphill the perception of the physical strenuousness is significantly lower for e-bike rides than for bike rides. Therefore it can be stated that e-bikes have the potential to motivate more people to ride a two-wheeler due to the support by electric propulsion.

However, the reported findings are derived from a relatively small dataset. Furthermore, we are fully aware that more than eight factors influence a two-wheeler's route choice. Studies in other cities are necessary to verify whether the observed street infrastructure preferences of bikers and e-bikers reflect actual preference differences or were influenced by the street layout in the area of the field study. Moreover, trip length differences between bikers and e-bikers should be considered by analyzing GPS tracks that are not tied to a specific origin-destination relation. The reported findings based on the rather qualitative questionnaire need to be verified by more quantitative tracking data.

As a next step an urban transportation mode choice model integrating e-bikes or a route choice model for e-bikers could be developed. Such an urban transportation mode choice model could shed light on the impact of e-bikes on the mode choice behavior of commuters. As a basis either data from a Stated Preference (SP) Survey or from a Revealed Preference (RP) Survey are necessary. SP surveys enable the analysis of a broad range of decision factors. However, participants of such surveys are only confronted with hypothetical decision situations. RP surveys however, investigate relevant decision factors based on a comparison of a real-world decision of a participant with actual alternatives. The major drawback of RP surveys is that only decision factors occurring on a participant's decision can be analyzed (Kroes and Sheldon 1988). Therefore, as discussed by Ben-Akiva et al. (1994), both types of survey data should be combined in order to benefit from the advantages of both approaches.

Acknowledgments We like to thank Moritz Meenen and Pratik Mukerji from the ETH spin-off company *ElectricFeel Mobility Systems GmbH* for supporting the field study with two-wheelers, tracking devices, and funding. The insightful comments from four anonymous reviewers helped to improve the final version.

References

Aultmann-Hall, L., Hall, F., & Baetz, B. (1997). Analysis of bicycle commuter routes using geographic information systems: Implications for bicycle planning. *Transportation Research Record: Journal of the Transportation Research Record, 1578*, 102–110.

Ben-Akiva, M., Bradley, B., Morikawa, T., Benjamin, J., Novak, T., Oppewal, H., et al. (1994). Combining revealed and stated preferences data. *Marketing Letters, 5*(4), 335–349.

Bohte, W., Maat, K., & Quak, W. A. (2008). A method for deriving trip destinations and modes for GPS-based travel surveys. In J. Van Schaick & S. C. Van der Spek (Eds.), *Urbanism on track* (pp. 129–145). Amsterdam, The Netherlands: IOS Press.

Brick, E., McCarthy, O. T., & Caulfield, B. (2012). Determining bicycle infrastructure preferences —a case study of Dublin. *Transportation Research Part D: Transport and Environment, 17*(5), 413–417.

Broach, J., Dill, J., & Gliebe, J. (2012). Where do cyclists ride? A bicycle route choice model developed with revealed preference GPS data. *Transportation Research Part A: Policy and Practice, 46*(10), 1730–1740.

Cellina, F., Förster, A., Rivola, D., Pampuri, L., Rudel, R., & Rizzoli, A. E. (2013). Using smartphones to profile mobility patterns in a living lab for the transition to e-mobility. In J. Hřebíček, G. Schimak, M. Kubásek, & A. E. Rizzoli (Eds.), *Environmental software systems. Fostering information sharing, IFIP advances in information and communication technology* (Vol. 413, pp. 154–163). Berlin, Germany: Springer.

Cherry, C., Worley, S., & Jordan, D. (2011). *Electric bike sharing—system requirements and operational concepts*. Paper presented at the 90th annual meeting of the Transportation Research Board, Washington, January 2011.

Chung, E.-H., & Shalaby, A. (2005). A trip reconstruction tool for GPS-based personal travel surveys. *Transportation Planning and Technology, 28*(5), 381–401.

Dalumpines, R., & Scott, D. M. (2011). GIS-based map-matching: Development and demonstration of a postprocessing map-matching algorithm for transportation research. In S. Geertman, W. Reinhardt, & F. Toppen (Eds.), *Advancing geoinformation science for a changing world* (pp. 101–120). Berlin, Germany: Springer.

De Hartog, J. J., Boogaard, H., Nijland, H., & Hoek, G. (2010). Do the health benefits of cycling outweigh the risks? *Environmental Health Perspectives, 118*(8), 1109–1116.

DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation, 12*(4), 41–56.

Dill, J. (2009). Bicycling for transportation and health: The role of infrastructure. *Journal of Public Health Policy, 30*, 95–110.

Dill, J., & Rose, G. (2012). Electric bikes and transportation policy: Insights from early adopters. *Transportation Research Record: Journal of the Transportation Research Board, 2314*, 1–6.

Fajans, J., & Curry, M. (2001). Why bicyclists hate stop signs. *Access, 18*(1), 28–31.

Golledge, R. G. (1995). *Defining the criteria used in path selection*. Paper presented at the international conference on activity based approaches: Activity scheduling and the analysis of activity patterns, Eindhoven, The Netherlands, May 1995.

Golledge, R. G., & Stimson, R. J. (1997). *Spatial behavior: A geographic perspective*. New York: Guilford Press.

Harvey, F. J., & Krizek, K. (2007). *Commuter bicyclist behavior and facility disruption*. St. Paul, Minnesota: Minnesota Department of Transportation, Research Services Section.

Heinen, E., Maat, K., & van Wee, B. (2013). The effect of work-related factors on the bicycle commute mode choice in the Netherlands. *Transportation, 40*(1), 23–43.

Heinen, E., van Wee, B., & Maat, K. (2010). Commuting by bicycle: An overview of the literature. *Transport Reviews: A Transnational Transdisciplinary Journal, 30*(1), 59–96.

Hunt, J. D., & Abraham, J. E. (2007). Influences on bicycle use. *Transportation, 43*(4), 453–470.

Kroes, E. P., & Sheldon, R. J. (1988). Stated preference methods. An introduction. *Journal of Transport Economics and Policy, 22*(1), 11–26.

Menghini, G., Carrasco, N., Schüssler, N., & Axhausen, K. W. (2010). Route choice of cyclists in Zürich. *Transportation Research Part A: Policy and Practice, 44*(9), 754–765.

Montello, D. R. (2005). Navigation. In P. Shah & A. Miyake (Eds.), *The Cambridge handbook of visuospatial thinking*. Cambridge: Cambridge University Press.

Parkes, S. D., Marsden, G., Shaheen, S. A., & Cohen, A. P. (2013). Understanding the diffusion of public bikesharing systems: Evidence from Europe and North America. *Journal of Transport Geography, 31*, 94–103.

Popovich, N., Gordon, E., Shao, Z., Xing, Y., Wang, Y., & Handy, S. (2014). Experiences of electric bicycle users in the Sacramento, California area. *Travel Behavior and Society, 1*(2), 37–44.

Rondinella, G., Fernandez-Heredia, A., & Monzon, A. (2012). Analysis of perceptions of utilitarian cycling by level of user experience. In *Proceedings of Transport Research Board Annual Meeting 2012*, Washington, DC.

Rose, G. (2012). E-bikes and urban transportation: Emerging issues and unresolved questions. *Transportation, 39*(1), 81–96.

Shaheen, S. A., Guzman, S., & Zuang, H. (2010). Bikesharing in Europe, the Americas, and Asia. *Transportation Research Record: Journal of the Transportation Research Board, 2143*, 159–167.

Stinson, M. A., & Bhat, C. R. (2003). Commuter bicyclist route choice: Analysis using a stated preference survey. *Transportation Research Record: Journal of the Transportation Research Board, 1828*, 107–115.

Stinson, M. A., & Bhat, C. R. (2005). *A comparison of route preferences of experienced and inexperienced bicycle commuters*. Paper presented at the 84th annual meeting of the Transportation Research Board, Washington, January 2005.

Stopher, P., Bullock, P., & Horst, F. (2002). Exploring the use of passive GPS devices to measure travel. In *Proceedings of the 7th International Conference on Application of Advanced Technologies in Transportation* (pp. 959–967). MA, USA: Cambridge.

Taylor, D., & Mahmassani, H. (1996). Analysis of stated preferences for intermodal bicycle-transit interfaces. *Transportation Research Record: Journal of the Transportation Research Board, 1556*, 86–95.

Tsui, A., & Shalaby, A. (2006). Enhanced system for link and mode identification for personal travel surveys based on global positioning systems. *Transportation Research Record: Journal of the Transportation Research Board, 1972*, 38–45.

United Nations. (2012). *World urbanization prospects: The 2011 revision*. New York: Department of Economic and Social Affairs, Population Division, United Nations. Available online: http://de.slideshare.net/undesa/wup2011-highlights. Accessed 15 November 2014.

Van der Spek, S. (2008). Tracking pedestrians in historic city centres using GPS. In S. Van der Spek, F. D. Van der Hoeven, & M. G. J. Smit (Eds.), *Street level desires* (pp. 86–111). Delft, The Netherlands: Delft University of Technology Urbanism.

Van der Spek, S., Van Schaick, J., De Bois, P., & De Haan, R. (2009). Sensing human activity: GPS tracking. *Sensors, 9*(4), 3033–3055.

Van Evert, H., Brög, W., & Erl, E. (2006). Survey design: The past, the present and the future. In P. Stopher & C. Stecher (Eds.), *Travel survey methods—quality and future directions* (pp. 75–93). Amsterdam, Oxford: Elsevier.

Weinert, J. X., Ma, C. T., Yang, X. M., & Cherry, C. (2008). The transition to electric bikes in China: Effect on travel behavior, mode shift, and user safety perceptions in a medium-sized city. *Transportation Research Record: Journal of the Transportation Research Board, 2038*, 62–68.

Wolf, J. (2006). Application of new technologies in travel surveys. In P. Stopher & C. Stecher (Eds.), *Travel survey methods—quality and future directions* (pp. 531–544). Amsterdam, Oxford: Elsevier.

Understanding Taxi Driving Behaviors from Movement Data

Linfang Ding, Hongchao Fan and Liqiu Meng

Abstract Understanding taxi mobility has significant social and economic impacts on the urban areas. The goal of this paper is to visualize and analyze the spatio-temporal driving patterns for two income-level groups, i.e. high-income and low-income taxis, when they are not occupied. Specifically, we differentiate the cruising and stationary states of non-occupied taxis and focus on the analysis of the mobility patterns of these two states. This work introduces an approach to detect the stationary spots from a large amount of non-occupied trajectory data. The visualization and analysis procedure comprises of mainly the visual analysis of the cruising trips and the stationary spots by integrating data mining and visualization techniques. Temporal patterns of the cruising trips and stationary spots of the two groups are compared based on the line charts and time graphs. A density-based spatial clustering approach is applied to cluster and aggregate the stationary spots. A variety of visualization methods, e.g. map, pie charts, and space-time cube views, are used to show the spatial and temporal distribution of the cruising centers and the clustered and aggregated stationary spots. The floating car data collected from about 2000 taxis in 47 days in Shanghai, China, is taken as the test dataset. The visual analytic results demonstrate that there are distinctive cruising and stationary driving behaviors between the high-income and low-income taxi groups.

Keywords Taxi driving behavior · Mobility pattern · Movement data

L. Ding (✉) · L. Meng
Department of Cartography, Technical University of Munich, Munich, Germany
e-mail: linfang.ding@tum.de

H. Fan
Department of GIScience, Heidelberg University, Heidelberg, Germany

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_13

1 Introduction

In modern society, advanced tracking technologies and devices, such as GPS receivers, mobile phones, have become increasingly pervasive, yielding massive movement data. This mobility data describes the changes of spatial positions of the mobile objects and provide the possibilities to investigate the urban dynamics. For instance, in intelligent transportation system, floating car data (FCD) collected at a relative low-cost can provide up-to-date high-quality traffic information (Huber et al. 1999). Some research works are dedicated to extracting geographic knowledge from movement data. Zheng and Zhou (2011) systematically investigated spatial trajectories from a wide spectrum of perspectives and disciplines, e.g. spatial database, mobile computing and data mining. Yuan et al. (2012a) presented a recommender system for both taxi drivers and taker using the knowledge of both passengers' mobility patterns and taxi drivers' picking-up/dropping-off behaviors learned from the GPS trajectories of taxicabs. In human geography, Liu et al. (2012a, b), Yuan et al. (2012b) examined large amounts of floating car data and mobile phone data respectively to understand human mobility patterns. In visual analytics, interactive visualization of movement data both at local scales focusing on individual trajectories (Guo et al. 2011; Tominski et al. 2012), or at large scales emphasizing on aggregated data (Andrienko and Andrienko 2011, 2013) have been comprehensively studied to extract significant traffic mobility patterns.

The goal of this paper is to visualize and analyze the spatiotemporal mobility/driving patterns of non-occupied taxis of two income-level groups, i.e. high-income and low-income taxis, inferred from the average daily income that can be derived from floating car data. Intuitively, taxi driving behavior can be different when the taxi is vacant or occupied, or when the taxi driver is experienced or novice. When the taxi is occupied, the taxi driver usually finds out the fastest route to send passengers to a destination based on his knowledge (Yuan et al. 2013). On the contrary, when the taxi is vacant, the taxi driver has a large freedom to plan his/her routing to minimize his/her waiting time of the next trip. Generally speaking, experienced taxi drivers are more likely to pick up their next passengers quickly while novice drivers may cruise on the roads for a longer time. Therefore, different taxi driving behaviors may exhibit significant influence on the income of taxi drivers.

Previous studies on the driving behavior such as Liu et al. (2009) categorize drivers into top driver and ordinary driver by their average daily income and conduct spatiotemporal analysis of their operation behavior and skill (as measured by income) based on the operation zone. Our research aims to provide a deeper understanding of driving behaviors between low- and high-earning taxis utilizing visual and computational methods. We distinguish top and bottom taxis and not only look into their cruising traces but also take into consideration of their stationary spots. Furthermore, the time span of the spatiotemporal FCD is longer and the amount of taxis for analyzing are more than the existing approaches. In the presented approach we analyze the taxi-driving behaviors based on: (a) their

incomes, (b) the spatiotemporal pattern of their cruising trips, and (c) the spatiotemporal pattern of their stationary spots. These analysis attempt to answer the questions such as what are the differences in terms of (1) the overall temporal patterns when they are cruising or stationary; (2) their spatial cruising distributions and (3) their stationary spatiotemporal characteristics.

The rest of this paper is organized as follows: in Sect. 2 the test dataset and preliminaries are described. In Sect. 3, we reconstruct the occupied trips and derive the taxi driver income, which is used to categorize taxis into high-income and low-income groups. Section 4 introduces an approach for the detection of the stationary spots. In Sects. 5 and 6, we investigate in detail the spatiotemporal mobility patterns of the cruising trips and the stationary spots for the two income-level taxi groups. In Sect. 7, we analyze and discuss the experiments of this paper. Finally Sect. 8 concludes this paper.

2 Test Dataset and Preliminaries

The test dataset are temporally ordered position records collected from about 2000 GPS-enabled taxis within 47 days from 10th May to 30th June 2010, in Shanghai. The temporal resolution of the dataset is 10 s and thus theoretically around 8000 GPS points of each car would be recorded in one day (24 h) given the GPS device effective. Each position record has nine attributes, i.e. car identification number, company name, current timestamp, current location (longitude, latitude), instantaneous velocity, and the GPS effectiveness. The detailed description of the fields is shown in Table 1.

In this work, two states of a taxi can be directly discerned based on the value of "car status", i.e. occupied (O) with a "car status" value of 1, and non-occupied (N) of 0. Figure 1 illustrates the raw GPS points of a taxi with an identification number of 10003 on the 12th May 2010 with the occupied and unoccupied states color coded in blue and red. In addition, we consider two sub-states of the non-occupied

Table 1 Description of the fields of floating car data

Field	Example field value	Field description
Date	20100517	8-digit number, yyyymmdd
Time	235903	6-digit number, HHMMSS
Company name	QS	2-digit letter
Car identifier	10003	5-digit number
Longitude	121.472038	Accurate to 6 decimal places, in degrees
Latitude	31.236135	Accurate to 6 decimal places, in degrees
Velocity	16.1	In km/h
Car status	1/0	1-occupied; 0-unoccupied
GPS effectiveness	1/0	1-GPS effective; 0-ineffective

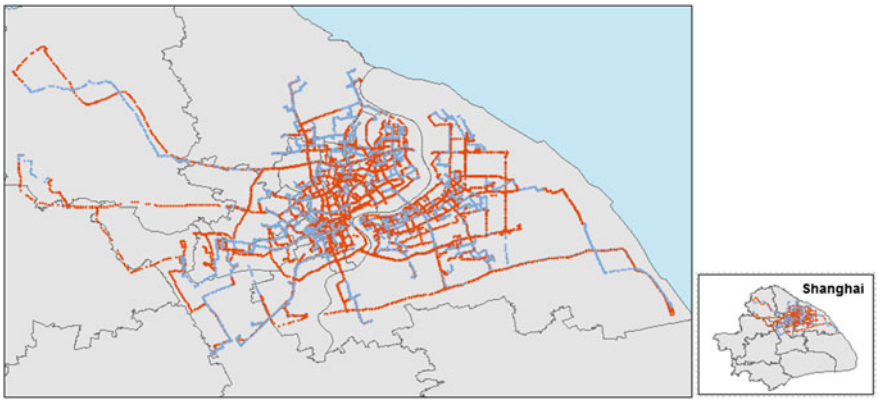


Fig. 1 GPS points of taxi with an ID of 10003 on the 12th May, 2010. *Red* and *blue dots* respectively indicate GPS points of occupied and unoccupied state

Table 2 Description of three states of a taxi

States	Sub states	Description
Occupied (O)		A taxi is occupied by a passenger
Non-occupied (N)	Cruising (C)	A taxi is moving without a passenger
	Stationary (S)	A taxi is static without a passenger

state, namely cruising (C) and stationary (S) state. Stationary state of a taxi means a taxi is vacant and static and refers to the GPS points with car status of 0 and instantaneous velocity (denoted by the "Velocity" attribute) of 0, while cruising state means a taxi is moving without a passenger (see Table 2).

This work adopts similar definitions of taxi trajectory and trip as in Yuan et al. (2012a) and defines a sequence of GPS points logged for a taxi as a taxi trajectory and a sub-trajectory with a single state as a taxi trip. According to the three states of a taxi defined in Table 2, this work introduces the occupied trips, cruising trips and stationary trips. Since a stationary trip comprises of a sequence of static GPS points at the same location and can be regarded as a stationary spot with associated stationary trip statistics, the following of this paper uses the term "stationary spot".

3 Derivation of the Income of Taxi Drivers

Occupied trips and non-occupied trips can be reconstructed by connecting the temporal sequences of GPS points with the "car status" attribute value of 1 and 0 respectively. A number of associated trip statistics, e.g., trip distance and duration,

Table 3 The taxi fare calculation system of Shanghai in 2010

	Item	Day timing (05:00 a.m.–23:00 p.m.)	Night timing (23:00 p.m.–05:00 a.m.)
$\mathbf{P}_0$	Minimum fare for first 3 km	12 CNY	16 CNY
$\mathbf{P}_3$	Fare above minimum fare until 10 km	2.4 CNY/km	3.1 CNY/km
$\mathbf{P}_{10}$	Fare above 10 km	3.6 CNY/km	4.7 CNY/km

are also derived. Based on the occupied trip distance and the taxi fare system of Shanghai in 2010 shown in Table 3, the average daily income of each taxi driver can be calculated. Formula (1) is used to calculate the fare of an occupied trip.

$$f(d) = P_0 + P_3 * \text{Min}(\text{Max}(d-3,0),7) + P_{10} * \text{Max}(d-10,0) \quad (1)$$

where d is the distance of the occupied trip.

In Fig. 2a, the histogram represents the distribution of the average daily income of the taxi dataset (2000 taxis in 47 days). The normal distribution of the daily income indicates the income discrepancy among the taxis. This paper categorizes the 100 highest-income taxis as the top group and the 100 lowest-income taxis as the bottom group. The following sections of this paper discuss the spatiotemporal analysis based on these two groups. Moreover, we estimate the occupied ratios by utilizing the distance of the occupied and unoccupied trip with the formula $dist(trips_{occupied})/(dist(trips_{occupied}) + dist(trips_{unoccupied})$. The line chart in Fig. 2b illustrates the occupied ratios of the top and the bottom taxis. Basically, the occupied ratio of both groups has several peaks (e.g. at 8 a.m.) and valleys (e.g. at 4 a.m.) and the high-income taxi group generally has a much higher occupied ratio. However, in the early morning between 2 a.m. and 7 a.m., the differences between the top and bottom taxi groups are much smaller than that in other time slots.

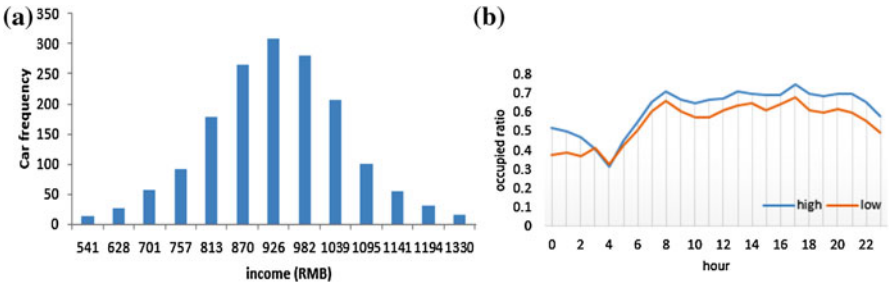


Fig. 2 **a** Average daily income distribution. **b** Occupied ratio of high- and low-income taxis

4 Detection of Stationary Spots and the Cruising Trips

This section details the process for detecting stationary spots from the non-occupied trips. A stationary spot is the location where a sequence of stop events occur and last for more than 5 min. A two-step approach is designed to detect the stationary spots. The specific workflow is illustrated in Fig. 3 and described as follows:

First, we aggregate a sequence of static GPS points ($v = 0$) of a taxi to predefined grids (grid size is about 100 m × 100 m) and calculate the centroids of the points inside each grid.

(1a) Extract raw GPS points from the dataset with cars of unoccupied state and with the instantaneous velocity of zero, and partition the study area into grids with the side length of l_g.

(1b) Assigning the extracted GPS points into the grids. In Fig. 3(1b), G1, G2, G3 and G4 are the grid divisions.

(1c) Calculate the duration and the count of the time-series GPS points inside each grid for each taxi. If a sequence of stop events in a grid lasts a long enough time period (e.g. more than 5 min) and are geographically close enough (e.g. $l_g/4$), we treat such sequence as a cluster. In Fig. 3(1c), the green dots in the grid G2 and G3 are two clusters of stop events.
Next, a refinement step is applied to the clusters, whose centroids are close to the boundary of the grid, by adjusting the surrounding grids to recalculate the clusters inside the new grid.

(2a) Move the grid by $l_g/2$ if the mean center inside a grid is close to its grid boundary. For instance, in Fig. 3(2a) the green rectangle named $G2_{shifted}$ is a new grid.

(2b) Repeat step (1c) and get the cluster in the new grid. The red dot in Fig. 3(2b) is the new mean center of $G2_{shifted}$.

(2c) Finally, the mean centers are identified as the stationary spots of the GPS points. In Fig. 3(2c), there are two stationary spots denoted by red dots.

The detection of stationary spots by the above procedure largely reduces the amount of GPS points. Note that the detected stationary spots are associated with temporal and thematic attributes (e.g. stationary duration, starting and ending timestamp) and they may refer to locations with different meanings such as parking lots, railway stations or hotels, or traffic congestion locations at road segments or intersections.

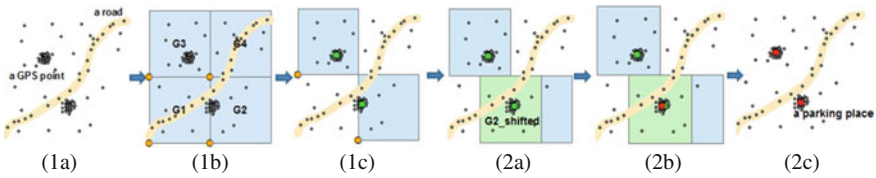


Fig. 3 The procedure of detecting the stationary spots. The *black dots* are stop events

Table 4 The numbers of the occupied, cruising trips and the stationary spots

	Top taxi group	Bottom taxi group
Occupied trips	217,253	110,753
Cruising trips	249,600	141,698
Stationary spots	30,623	24,263

After the detection, the stationary spots can be excluded from the non-occupied trips and the rest of the GPS points are of the cruising state and can be used for reconstructing the cruising trips. Table 4 shows the numbers of the reconstructed occupied, cruising trips and the stationary spots analyzed in this paper.

5 Spatiotemporal Patterns of the Cruising Trips

When a taxi driver is cruising, he or she plans to minimize his or her cruising time and intends to quickly pick up the next passenger. To investigate the temporal patterns of the cruising trips between top and bottom taxis, we calculate the average hourly cruising duration (shown in Fig. 4) between the top and bottom taxi groups. The top taxis are normally cruising longer than the bottom taxis, especially in the time slots of midnight and early morning from 22 p.m. to 6 a.m. During the day, typically at 9 a.m. and 17 p.m., the differences of their cruising durations are relatively small.

To get an overview of the spatial distribution of the cruising trips, we examine the distribution of the mean centers of the cruising trips.

Given a taxi trip $t = ((x_1, y_1), \ldots, (x_n, y_n))$, its spatial mean center is

$$mc_{trip}(t) = (1/n \sum_{i=1,..n} x_i, 1/n \sum_{i=1,..n} y_i)$$

Consequently, given a set of taxi trips $T = (t_1, \ldots, t_m)$ in one day, the spatial mean center of the trips is

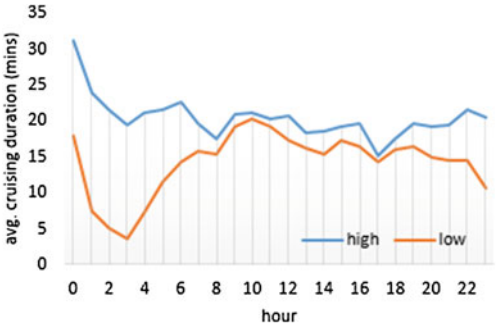


Fig. 4 The average hourly cruising duration for top and bottom drivers

$$mc_{trip\_day} = \frac{1}{m}\sum_{t\in T} mc_{trip}(t)$$

Based on the above definitions, the average daily spatial mean centers of each car in the 47 days are calculated. Furthermore, to investigate the amount of variation or dispersion for each car, we also calculate the standard deviation of the daily spatial mean centers of each car.

Figure 5 shows the distribution of the average daily spatial mean centers (Fig. 5a, b) of the cruising trips and their standard deviations (Fig. 5c, d) of the bottom (Fig. 5a, c) and top (Fig. 5b, d) taxis. The rings in Fig. 5a, b are spatial partitions indicating the areas from the center of Shanghai to the peripheral. One can easily observe that the spatial distribution of the average daily cruise centers of the top taxi group is more compact and concentrated in the city center, while the spatial distribution of the daily cruise mean centers of the bottom taxi group is more dispersed. The orange ellipses in Fig. 5c, d indicate that the daily spatial mean center for the top taxi group is more centralized than those of the bottom taxi group. For the bottom taxis, the further they cruise from the city center, the more they spread from their daily cruising spatial mean centers.

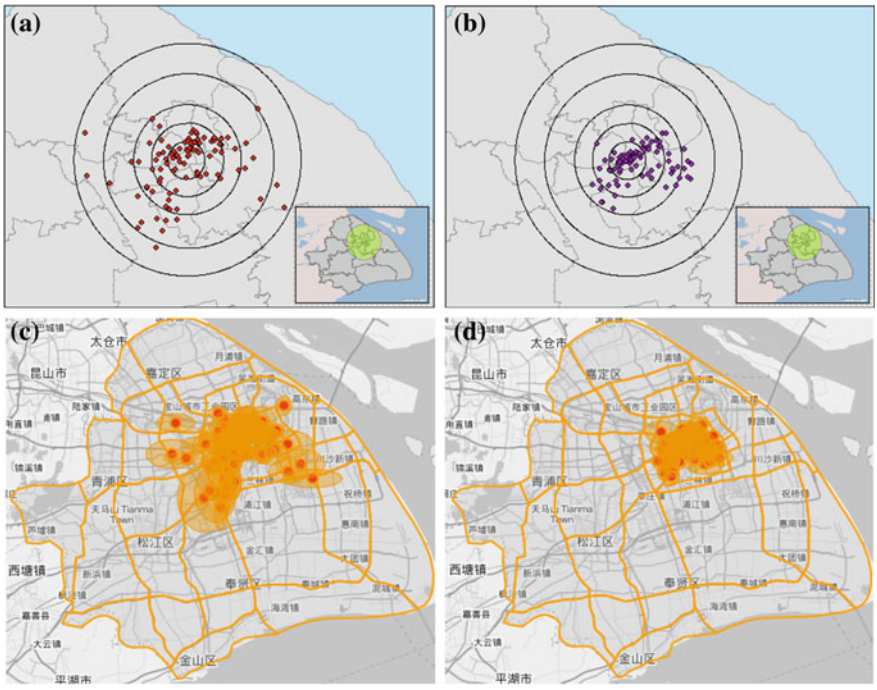


Fig. 5 The average daily spatial mean centers of the cruising trips (**a**, **b**) and their standard deviations (**c**, **d**) of the bottom (**a**, **c**) and top (**b**, **d**) taxi groups. In **a**, **b**, *dots* represent the average daily spatial mean centers of each taxi and *ellipses* in **c**, **d** their standard deviations

6 Spatiotemporal Patterns of the Stationary Spots

Recall that in this paper, stationary spots are locations where taxis are static for more than 5 min and they may refer to parking lots, taxi stands or road segments where the traffic congestions happen. In this section, we investigate the spatio-temporal distributions of the stationary spots. Firstly Sect. 6.1 inspects the general temporal distribution of the stationary durations. Next Sects. 6.2 and 6.3 study more detailed spatiotemporal patterns of stationary spots of long break and in off-peak hours.

6.1 General Temporal Patterns

To get a temporal overview of the stationary spots, we calculated the average stationary durations of all the 2000 taxis, divided into 15-min time intervals. The average stationary duration is visualized in a time graph (see Fig. 6a). The rows represent the dates from 10th May to 30th June, and the columns represent every 15-min in 24 h. The color scheme is chosen from the Color Brewer system. The darker the individual block is, the longer the taxis remain static. The white color means missing data (11th May, 4 hours on 11th June, and 12th–14th June).

The time graph in Fig. 6a shows significant daily and weekly patterns. The daily pattern is illustrated in each row where the dark red colors occur at the early morning (1:00 a.m.–6:00 a.m.), noon (especially from 11:30 a.m.–13:00 p.m.), and in a short period at evening (around 19:00 p.m.). These time slots can be interpreted as the sleeping, lunch and dinner periods respectively.

The weekly pattern is that for every seven rows the dark red color occurrences shifted about 2 or 3 h right, which means that stationary durations at weekends are about 1–2 h longer and later than that at weekdays. Interestingly, one exceptional weekend pattern can be found on 15th and 16th June (Tuesday and Wednesday).

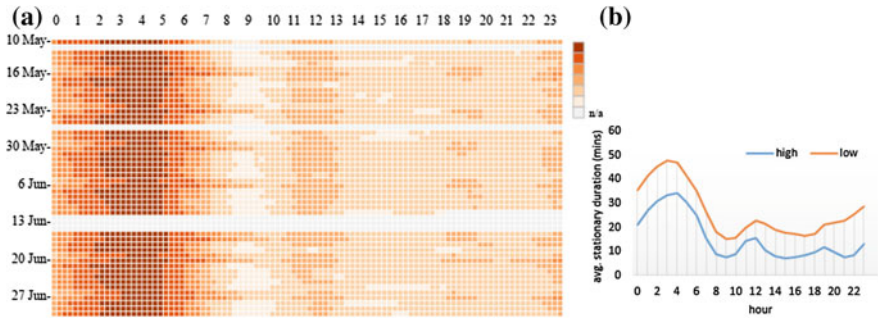


Fig. 6 **a** The time graph of the stationary durations aggregated into 15-min intervals, **b** the average hourly duration at the stationary spots for top and bottom drivers

By looking up the Chinese calendar of 2010, we found that 14th–16th June (unfortunately the data on 14th June are missing) is the 3-day national holiday for the Duanwu Festival.

To study the difference of the top and bottom taxi groups, we also inspected the time graphs of the stationary duration for them and found similar daily and weekly patterns. Next, we created the line graph of Fig. 6b which shows the average hourly stationary duration of the top and bottom drivers. Unsurprisingly, the bottom taxi drivers stay at one places longer than the top taxi drivers.

6.2 Spatiotemporal Patterns of Long Break Stationary Spots

In this subsection, we investigate the spatiotemporal distribution of stationary spots with relatively long durations, which are normally breaks or long waiting periods. For instance, during the night, such stationary spots are mostly related to sleeping, and during the day, they may imply long waiting in a queue. It is important to know the spatiotemporal distribution of such events since they may have significant impacts on the taxis' overall income.

Here, we extract and examine the stationary spots with the duration more than 30 min. A density-based spatial clustering approach is applied to cluster these stationary spots. Specifically, DBSCAN (density-based spatial clustering of applications with noise) algorithm (Ester et al. 1996) is applied to detect clusters of arbitrary shapes, with the distance parameter set to 20 m and the minimum point parameter to 10. The number of the extracted spatial clusters of the bottom and top taxi groups are 216 and 137 respectively.

Furthermore, we divided one day into four time intervals, namely midnight (12 a.m.–06 a.m.), morning (06 a.m.–12 p.m.), afternoon (12 p.m.–18 p.m.) and evening (18 p.m.–12 a.m.) and calculate in each cluster the number of cluster elements (stationary spots) in each interval. Pie chart maps are used to visualize the spatial and temporal distribution of the clusters. The location of each pie chart represents a cluster centroid; the size of the pie chart is proportional to the total number of elements in each cluster; and the colors (i.e. orange, red, yellow and blue) of the pie chart sectors correspond to the four time intervals and these sectors are proportionally sized to the respective number of the cluster elements in each time interval.

The two screenshots in Fig. 7 show the spatiotemporal distribution of the cluster centroids of stationary spots with the duration more than 30 min. There are two relatively large pie charts in both graphs, which means that both top and bottom taxis often stay at these two places. These two pie charts have large proportions of afternoon and evening. Actually, they are located at the two transport hubs of Hongqiao (left) and Pudong international airport (right). We also notice that the cluster at Pudong for top taxi groups is much smaller than that for the bottom taxi group. For both top and bottom taxi groups, the relative small pie charts are mostly in orange and correspond to the midnight while for the bottom taxi groups there are

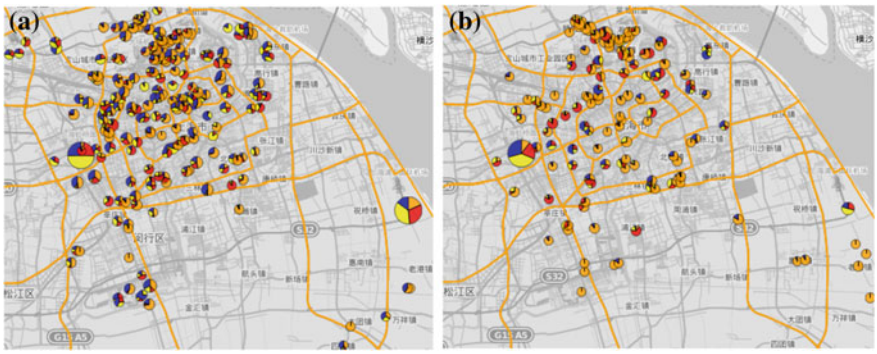


Fig. 7 Comparison of the spatiotemporal distribution of the cluster centroids of the stationary spots between the bottom (**a**) and top (**b**) taxi drivers. The predefined four time slots are color coded in *orange*, *red*, *yellow* and *blue*; the number of stationary spots in the each cluster is proportionally sized to the size of each pie

more blue portions (Fig. 7a) and refers to more stationary spots during the evening interval (18 p.m.–24 p.m.). We can interpret that the small circles with a large midnight and evening portions may correspond to their long-break and familiar places, e.g. the locations of the taxi companies or taxi drivers' home.

6.3 Spatiotemporal Patterns of Stationary Spots in Off-Peak Hours

We noticed in Fig. 2b there is a large discrepancy of the occupied ratios between the top and bottom driver groups from 10 a.m. to 12 p.m. This off-peak-hour occupied ratio difference might result from distinct driving behaviors and is interesting to get an insight.

To understand the spatiotemporal patterns of the stationary spots from 10 a.m. to 12 p.m., we extract from the detected stationary spots about 1100 and 1500 stationary spots during this time interval for the bottom and top taxi groups respectively.

Until now, we did not differentiate the meanings of the stationary spots. Here, we classify the stationary spots from 10 a.m. to 12 p.m. to traffic congestion and parking places. To identify the traffic congestion clusters, we utilize the road networks to calculate the distance from the clustered stationary spots to their nearest roads. A threshold (e.g. 20 m) is adopted between a traffic congestion place (<20 m) and a parking place (≥20 m). Figure 8 shows the space-time cube of the stationary spots of the bottom and top taxi groups from 10 a.m. to 12 p.m. in the 47 days. Red dots represent the traffic congestion places and yellow dots the parking places. From Fig. 8, we can see that there are obviously more parking events indicated by the

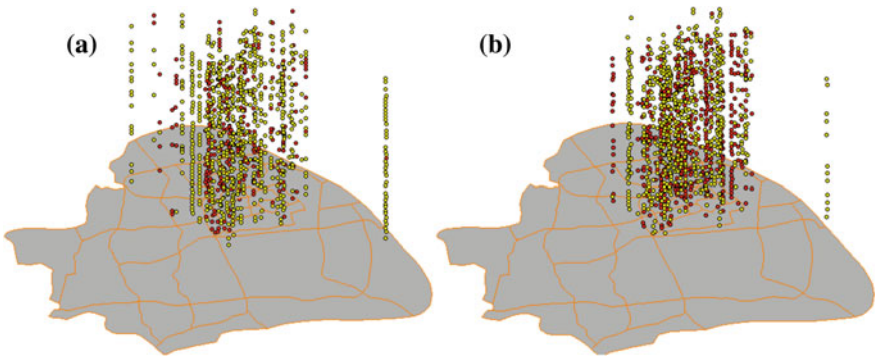


Fig. 8 Comparison of the spatiotemporal distribution of the stationary spots with space-time cube between the bottom (**a**) and top (**b**) taxi groups from 10 a.m. to 12 p.m. in the 47 days. *Red dots* represent the traffic congestion places and *yellow dots* the parking places

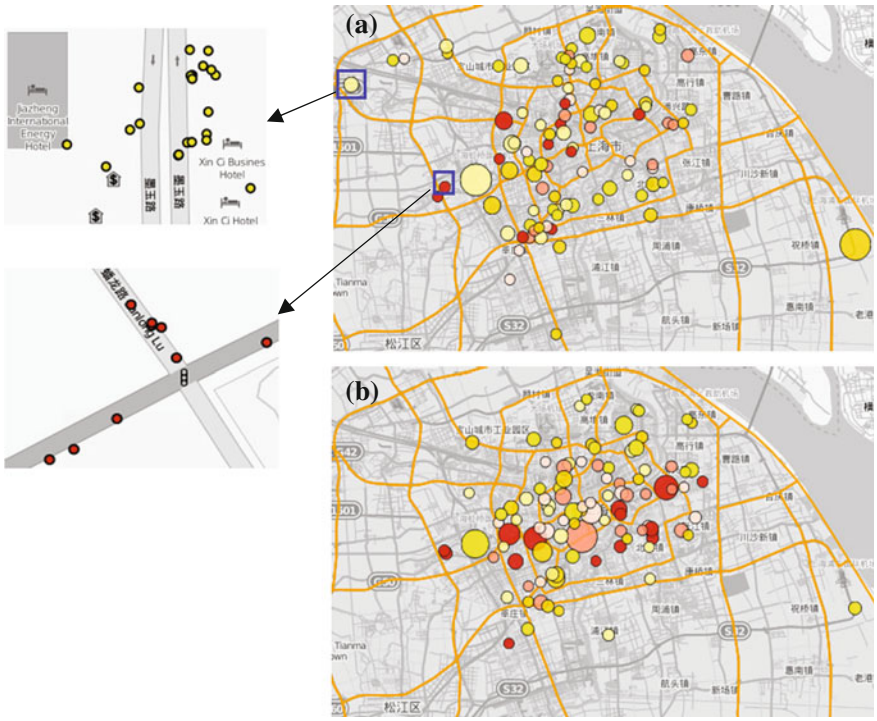


Fig. 9 Comparison of the spatiotemporal distribution of the parking and traffic congestion places between the bottom (**a**) and top (**b**) taxi groups. The parking places are in *yellow* and the congestion places in *red*. The color values correspond to the average duration at the places. The size of the *circle* is proportional to the count of the cluster elements of parking or traffic congestion

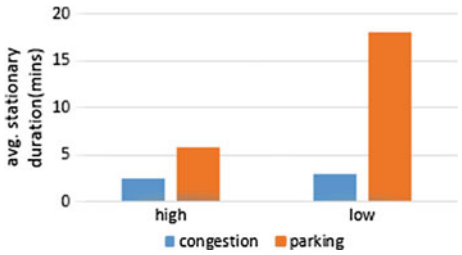


Fig. 10 The average stationary duration spent on the traffic congestion and parking places (waiting for passengers) for top and bottom taxi groups

yellow dots from the bottom taxi group (Fig. 8a), while there are more congestion events denoted by the red dots from the top taxi group (Fig. 8b).

To compensate the occlusion effect the space-time cube, a pie chart map of the cluster centroids are inspected (shown in Fig. 9). Similarly, we apply the DBSCAN clustering method and get 95 and 102 clusters after the clustering method. Each pie chart corresponds to a cluster centroid. If more than half of the elements in a cluster are traffic congestion, the cluster centroid is regarded as a traffic congestion place and coded in red; otherwise it is a parking place and coded in yellow. The color values correspond to the average hourly stationary duration in the cluster. The size of the pie charts is proportional to the number of elements in clusters. There are about 33 cluster centroids representing traffic congestions for the bottom taxis and 47 for the top taxis. We can see from Fig. 9 that bottom taxi group are more likely go eastward to the airport "Pudong" at this time interval with a longer parking time.

Figure 10 shows the average stationary duration (in minutes) spent on the traffic congestion and the parking places (waiting for passengers) for top and bottom taxi groups. The average time spend on traffic congestion is more or less the same, while the parking time from 10 a.m. to 12 p.m. for the bottom taxi group are much longer than the top taxi group.

7 Analysis and Discussion

The above sections studied in detail the spatial and temporal distribution of the trajectory traces of top and bottom taxi groups, which reflects their distinctive driving behaviors. Now we try to answer the questions proposed in the introduction for top and bottom taxi groups.

Differences of the overall temporal patterns when they are cruising or stationary The trend of the occupied ratios between the top and bottom taxi groups (Fig. 2b) reveals that the top taxi group drives longer distance with passengers than the bottom group. In spite of their longer occupied trips, in terms of cruising durations, the top taxi group normally cruises longer than the bottom taxi group especially in the evening and during the midnight (see Fig. 4), which might reduce the profit via

the cost of gas consumptions. However, in terms of stationary durations, top taxis have constantly shorter average durations (see Fig. 6b), which might in turn compensate the loss of longer cruising. In addition, we also found the daily and weekly routines of all the taxis by investigating their stationary durations (see Fig. 6a). The daily patterns show that long breaks often occur in the midnight and early morning, lunch or dinnertime. The weekly pattern shows obvious differences between weekdays and weekends.

Differences of their spatial cruising distributions In terms of spatial driving behaviors, the cruising patterns of the top taxi group are more compact and concentrated in the city center, as shown in Fig. 5, while the spatial distribution of the bottom taxi group is more dispersed and the further they cruise from the city center, the more they spread. Moreover, comparing the cruising patterns in Fig. 5 with the spatial distribution of the long break stationary spots, we can observe that there are no obvious relationships of the cruising mean centers and the long-parking places around midnight, and thus can interpret that taxis might not cruise around their long-parking or long-break places.

Differences of their stationary spatiotemporal characteristics We observe from the occupied ratio plot (see Fig. 2b) that there are relatively large differences during off-peak hours, which might be the main reason that result in the income differences. Thus we study in detail the spatial and temporal distributions during the off-peak time intervals, especially from 10 a.m. to 12 p.m., and found that the top and bottom taxi group spend similar quantity of time on the road congestions but significantly distinct quantity of time on the parking places (see Fig. 10). One reason might be that bottom drivers wait much longer in some places, for instance, in Fig. 9, we can easily see that there are a large number of bottom taxis (Fig. 9b) in the Pudong airport waiting for a relative long time period.

8 Conclusion

In this paper, we investigated the spatio-temporal patterns of the driving behaviors between low- and high-income taxis from large amount of floating car data in Shanghai. This work firstly differentiates two income-level driver groups derived from the GPS-enabled taxi trajectory data. An approach is designed to detect the stationary spots from massive trajectory data. To investigate the cruising patterns, we calculate and visualize the average daily mean centers of the cruising trips as well as the variation of the daily cruising mean centers. With regard to the stationary spots, we design a time graph to show the temporal patterns based on the aggregated durations of the stationary spots. Based on the clustering result and a distance method, parking places and traffic congestion are distinguished. A space-time cube is used to show the clustered spatiotemporal pattern of the traffic congestions and the parking places and a pie chart map to reveal the aggregated cluster centroid.

Preliminary results show that there are obvious driving behavior differences between the low-income and high-income taxis. The top taxi group mostly cruises

in the city center while the bottom taxi group tends to cruise in the peripheral areas of Shanghai. The evidences can be found by the spatial distribution of their respective cruising mean centers and the spatial variation of the daily cruising mean centers. With regard to the stationary state, both top and bottom taxi groups exhibit a similar daily and weekly temporal pattern. However, compared to the top taxi group, the bottom taxi group generally has a much longer waiting time.

This paper utilizes various visualization and computational methods to discover knowledge hidden in the massive floating car data. For instance, a grid-based approach is designed for the detection of the stationary places and a density-based clustering approach for clustering the detected stationary places. Line charts and time graphs are used to reveal the temporal patterns; pie chart maps and space-time cubes are applied to reveal the spatial patterns. In the future, further visualization options will be experimented and tested with target users.

Acknowledgments This work is partially funded by the China Scholarship Council. We would like to thank Prof. Chun Liu (School of Surveying and Geoinformatics, Tongji University) for sharing with us the Shanghai Taxi FCD dataset.

References

Andrienko, N., & Andrienko, G. (2011). Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics (TVCG), 17*(2), 205–219.

Andrienko, G., Andrienko, N., Hurter, C., Rinzivillo, S., & Wrobel, S. (2013). Scalable analysis of movement data for extracting and exploring significant places. *IEEE Transactions on Visualization and Computer Graphics, 19*(7), 1078–1094.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, U. Fayyad & M. Usama (Eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 226–231, ISBN 1–57735-004-9). Palo Alto: AAAI Press.

Guo, H., Wang, Z., Yu, B., Zhao, H., & Yuan, X. (2011). TripVista: triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *Proceedings of IEEE Pacific Visualization Symposium (PacificVis 2011)* (pp. 163–170), Hong Kong, March 1–4, 2011.

Huber, W., Lädke, M., & Ogger, R. (1999). Extended floating-car data for the acquisition of traffic information. In *Proceedings of the 6th World Congress on Intelligent Transport Systems*.

Liu, L., Andris, C., Biderman, A., & Ratti, C. (2009). Uncovering taxi driver's mobility intelligence through his trace. *IEEE Pervasive Computing*.

Liu, Y., Kang, C., Gao, S., Xiao, Y., & Tian, Y. (2012a). Understanding intra-urban trip patterns from taxi trajectory data. *Journal of Geographical Systems, 14*(4), 463–483. doi:10.1007/s10109-012-0166-z.

Liu, Y., Wang, F., Xiao, Y., & Gao, S. (2012b). Urban land uses and traffic 'source-sink areas': Evidence from GPS-enabled taxi data in Shanghai. *Landscape and Urban Planning, 106*(1), 73–87. doi:10.1016/j.landurbplan.2012.02.012.

Tominski, C., Schumann, H., Andrienko, G., & Andrienko, N. (2012). Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics, 18*(12), 2012.

Yuan, J., Zheng, Y., Sun, G., & Xie, X. (2013). T-drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE Transactions on Knowledge and Data Engineering (TKDE), 25*(1), 220–232.

Yuan, J., Zheng, Y., Zhang, L., & Xie, X. (2012a), T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Yuan, Y., Raubal, M., & Liu, Y. (2012b). Correlating mobile phone usage and travel behavior—A case study of Harbin, China. *Computers, Environment and Urban Systems, 36*(2), 118–130.

Zheng, Y., & Zhou, X. (2011). *Computing with spatial trajectories*. Berlin: Springer. ISBN:978-1-4614-1628-9.

Automated Generation of Indoor Accessibility Information for Mobility-Impaired Individuals

Nemanja Kostic and Simon Scheider

Abstract One important issue in developing assistive navigation systems for people with disability is the accuracy and relevancy of the systems' knowledge bases from the perspective of these special user groups. The theory of affordances coupled with computer-based simulation offers a solution for automating the extraction of the relevant information from readily available sources—architectural floor plans. Simulation of movement in a wheelchair can be used to compute the accessible space of an indoor environment by comparing the degree of match between geometrical demands of navigation and the relevant physical properties of the environment. We also investigate what constitutes the right level of representation of the environment and adopt the grid graph model as suitable both for accessibility computation and for deriving higher-level networks of places and their connections that facilitate orientation and user-system interaction.

Keywords Building accessibility · Affordance simulation · Grid graph · User relative navigation support · People with disability

1 Introduction and Motivation

The spatial decision-making process of people with disability can be facilitated by building assistive navigation systems offering access to environmental information that would otherwise either be out of reach or acquirable at the expense of much

N. Kostic (✉)
Institute for Geoinformatics, University of Münster,
Heisenbergstr. 2, 48149 Münster, Germany
e-mail: n_kost01@uni-muenster.de

S. Scheider
Institut für Kartografie und Geoinformatik, ETH Zürich,
Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland
e-mail: simonscheider@web.de

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_14

effort and frustration. Historically, such navigation systems have overwhelmingly dealt with outdoor environments since precise localisation techniques (such as GPS) were only available outside. However, recent developments in sensor-based positioning and ubiquitous computing have extended these efforts indoors (Afyouni et al. 2010). Indoor spaces pose their own unique obstacles for attempts to develop successful navigation aids; one such challenge is the availability of navigationally relevant information, especially for architecturally complex buildings.

The type of disability—motor, visual, auditory, cognitive—plays a crucial role in determining the precise contents of an assistive system's knowledge base. To focus our research, we consider an example scenario of a wheelchair user planning a visit to a large public building. Here, the most critical piece of information concerns accessibility: to what extent the indoor environment allows the user to move between indoor locations of interest, and what are the optimal ways to do so.

When information on accessibility is not readily available, we should find a way to obtain it in an exhaustive, reliable and relatively simple way. A common way to assess the accessibility of an environment is to do a survey while making note of potential obstacles for movement (presence of stairs/ramps, ramp slopes, etc.). This raw data can be compared to established criteria for accessibility assessment [such as legally defined guidelines for the design of public buildings; see for example (Neufert 2005)]; in this way, accessibility is determined in situ and later entered as annotation into a computer model of the environment for use in navigation systems or further analyses. The process is relatively easy to perform (though not always to manage), but also time-consuming and marked by uncertainty: it cannot guarantee exhaustiveness and is often dependent on subjective assessment.

Automation of accessibility assessment has been suggested in research on public transportation planning (Jonietz et al. 2013; Jonietz and Timpf 2013) and ergonomics in the workplace (Budziszewski et al. 2011); how to go about this task in the context of assistive indoor navigation systems is the focus of this paper. We propose a method to compute the user-relative accessible space of an indoor environment as well as a set of accessible paths between indoor places. We also discuss the right level of representation of the indoor environment to serve as input for the computation, and opt for one that preserves detailed geometrical properties of the environment while simultaneously allowing a straightforward extraction of a network of accessible places and their connecting paths.

The next section presents previous work related to our goal, before we turn to the specifics of the proposed methodology of accessibility information extraction, a use case of a real-world indoor environment and finally a discussion of open questions and possible ways of integrating the resulting information into different types of navigation systems.

2 Discussion of Related Work

2.1 Affordances

Each kind of disability has its own distinctive effects on navigation in space. One valuable attempt to theoretically model this interdependence of individuals' capabilities and the environment they act in has been made with the theory of affordances. Gibson introduced the idea of *affordance* as opportunity for action offered by the environment: different objects or their constellations are suitable for different types of use, and humans and other animals can perceive and act on these opportunities (Gibson 1977). In this way, the environment takes on meaning, in the sense that it carries information that can guide behaviour (Scarantino 2003). This role of affordances in cognitively modelling human navigation has been discussed, modelled and tested using computer simulation in Raubal (2001a, b), Raubal and Worboys (1999). To say that there is an interdependence between these action potentialities and the acting entities means that an affordance is what it is only in relation to a specific kind of agent. A person in a wheelchair faced with a flight of stairs "perceives obstacles where other people just perceive a step they can climb" (Ortmann and Kuhn 2010). For any customised information system, modelling the environment in terms of affordances offers "an experiential view of space, because they offer a user-centred perspective" (Raubal 2008).

2.2 Situated Simulation

What determines the existence of an affordance? The agent-environment complementarity that the notion of affordance models entails that an affordance can be assigned to an object only when a potential action exists that includes the object (Scarantino 2003). Recent developments in cognitive science hint in the direction of action-dependent meaning as well. Barsalou refers to ad-hoc grouping of environmental objects based on their usefulness for the action being planned or executed; in this way dynamic categories arise, such as 'things to stand on' when one is working out a strategy for replacing a lightbulb on the ceiling. This judgement of usefulness of an object depends on mentally performing—i.e. simulating—the action on the object and relies on the object's affordances being encoded in our concept of the object: concepts are toolboxes for action (Barsalou 1999, 2003). Building on this, Scheider (2011) proposed to ground affordances as perceivable potential events—successful simulations of actions generated while processing environmental input.

Experiments have shown that not only are people very good at correctly judging objects in this way, but that this process can be quantified as well. Warren's trials resulted in a numerical value that determined the existence of the affordance of *climbability* of a flight of stairs: a person will perceive the stairs as affording the action of climbing if the ratio between his/her leg length and the stairs' riser height

does not exceed 0.88 (Warren 1984). Except for simple cases, however, body scales are not enough to solely explain the perception of affordances, because they cannot fully capture what one can do—one's abilities, or functional properties (Chemero 2003). A paralysed individual's leg length may be the same as that of an able-bodied person but the environment's climb ability affordances are drastically different for the two of them.

Two main insights of the work on the emergence of affordances are relevant for our goal. Firstly, treating affordances as agent-action-environment relations means that a record of properties of an environment cannot be equated with a description of its affordances—these emerge only when concrete actors and actions enter into the equation. Furthermore, the set of affordances of a certain environment for a certain action can be derived by situated simulation of the action, where situatedness refers to its grounding in concrete agent-side constraints, and simulation to the possibility of determining the affordances independently of the action being actually performed.

2.3 Automated Mobility Affordance Assessment

The idea of automating the task of obtaining environmental information meaningful for navigation appears in Jonietz and Timpf (2013); it is further elaborated in Jonietz et al. (2013). The authors propose a computer-based "translation of selected environmental attributes [of public park paths] into a scaled suitability value for individual mobility" as an alternative to subjective or rule-of-thumb affordance determination. *Suitability* refers here to an extension of the concept of affordance beyond simple (im)possibility of action to include different levels to which an action can be afforded by an object. For example, a ramp may in principle afford movement to many people but will demand different amounts of effort from each of them, which may significantly influence their spatial decision making.

In a related application area, computer simulation was suggested in Budziszewski et al. (2011) to assess the ergonomic quality of workplaces using 3D virtual reality techniques. A wheelchair user was modelled based on statistical data on maximal arm reach to identify zones out of reach of the user and assess the need for a rearrangement of the work environment. A similar goal drove the development of the HADRIAN (http://www.lboro.ac.uk/microsites/lds/sammie/reshad.htm) database and SAMMIE simulation environment; these encompass not only the specific needs of disabled individuals but also the effects of age and/or difference in body scales, thereby allowing increasingly fine adjustments of environments to individual needs.

2.3.1 The Environment

To determine the degree to which an environment affords basic mobility—the environment's accessibility—we must focus on those of its features that enable or

impede movement; when the task is being automated, we are immediately faced with the question of choice of an adequate input spatial model, since this will determine the amount and type of environmental features on which to run the analysis. Jonietz et al. (2013), Jonietz and Timpf (2013) opt for a network model that represents park paths as edges and their intersections as nodes, with navigationally relevant properties such as length or slope—and consequently the resulting suitability values—aggregated on path level.

When we deal with an environment where movement is restricted to clearly delineated objects such as paths in a park, this aggregated approach is justified: mobility affordances can be understood as properties of individual paths. However, in indoor environments such as halls or rooms there are not always obvious paths: such spaces appear not to be discretised into networks but rather exhibit continuity and are better described as scenes (Rüetschi and Timpf 2005). As noted in Swobodzinski and Raubal (2008), pedestrians are in general not constrained to linear routes like vehicles are: to exhaustively model indoor movement using a network we would have to identify all possible paths between all possible pairs of destinations, ending up with huge networks even for moderately complex indoor environments.

It is clear then that aggregating environmental properties into discrete objects is not an optimal solution for indoor accessibility assessment. If one can move between two locations in a room, which of the many possible paths between them is the affordance bearer? Selecting one of them arbitrarily would imply unjustified 'gerrymandering' [to borrow a phrase from Lewis as quoted in Scarantino (2003)], whereas modelling each one explicitly is very difficult. In an indoor environment, mobility affordances are better thought of as attributes of the continuous space itself, and only after accessibility has been assessed on the level of the continuous geometry of obstacles and free space can we start breaking down the environment into destinations (which is quite arbitrary below the level of obvious architectural units such as rooms or corridors) and paths between them. For accessibility analysis, therefore, we need an input spatial model preserving the continuity of space.

2.3.2 The Agent and Its Activity

The other side of the affordance relation comprises agent-side properties; which among these are relevant in determining environmental affordances depends on the action in question. As we have seen, body scales are just a stand-in for what one can effectively do, and not always a good one at that; is there another way to encode ability? The procedure presented in Jonietz et al. (2013), Jonietz and Timpf (2013) builds on the idea of affordances as ratio values as outlined in Sect. 2.2: different levels of various factors of motor ability are expressed numerically and then set against the corresponding physical properties of the environment in order to quantify suitability.

Another way to quantify motor ability is hinted at in Afyouni et al. (2010). The authors propose a spatial model for indoor navigation that consists of three levels:

spatial, feature and action. Whereas the spatial level captures continuous geometric information on indoor environments, the feature level explicitly models objects (mobile and static). One important set of attributes of an object consists of its interaction spaces, which capture the different spatial extents needed to perform actions that include the object ("operational space"). We can use such interaction spaces as a way to derive mobility affordances by simulating actions.

2.4 Space and Place in Indoor Navigation Systems

The basic functions of indoor navigation systems comprise user localisation, path planning, directions derivation and provision of information on surrounding objects (Fallah et al. 2013). These services rely on two broad categories of spatial knowledge encoded in navigation systems: geometric and semantic, where the former models space as a continuous field while the latter decomposes space into places—chunks of space to which human-readable descriptions are attached (e.g. "You are in *the entrance hall*."), making it better suited to user-system interaction (Afyouni et al. 2012; Richter et al. 2011). As seen, spatial geometry determines some environmental affordances; how affordances may combine to determine the place structure of space is discussed in Scheider and Janowicz (2014).

Looking back at our hypothetical scenario, what is sought is information on the extent to which the building's space offers basic mobility to the user, while asking for answers in terms of the building's places (e.g. "Can I—and what is the easiest way to—get from the entrance hall to room 3?"). This twofold perspective on environments—space versus place, geometry versus semantics (also: continuity vs. topology, Li et al. 2010)—is an integral part of accessibility assessment when its final goal is its use in navigation systems. It appears that there are conflicting demands on the input model for our procedure: it should be both non-network (continuous) and network (discrete/place-based). A model that captures environmental properties in a way that fits both descriptions is discussed in Sect. 3.2.

3 Discussion of Methodology

In this section we propose a methodology to automatically extract accessibility information on indoor environments for use in assistive navigation systems, focusing on the case of wheelchair users. We start with a widespread and readily available information source on indoor spaces—architectural floor plans in CAD format—as a sufficient record of environmental properties for accessibility computation. Integrating the insights presented in Jonietz et al. (2013), Jonietz and Timpf (2013), Budziszewski et al. (2011) and Afyouni et al. (2010), we propose to derive an indoor environment's mobility affordances for a wheelchair user by simulating his/her movement in space. This is done by matching the geometrical

constraints of actions involved in moving in a wheelchair with the geometry of obstacles in the environment. Proceeding from this accessibility assessment at the level of environmental geometry we then derive a set of optimal paths between places in the environment. The way we conceptualise agents, their movement and the environment is explained in the following subsections, before we outline the procedure itself.

3.1 Modelling the Agent and Action

We propose an algorithm for mobility affordance derivation that performs an exhaustive analysis of an indoor environment for the possibility of movement. To do this, we discretise continuous path taking into a set of moments; at each moment the moving agent occupies a particular location in space while performing what we term a *movement primitive*: the agent either simply fits into the space—only to move at the next moment to the adjacent location in the same direction—or takes a turn in order to change direction.

At each (non-obstacle) location, then, we test for two conditions: first, with the agent's centroid at the location, whether the surrounding space affords simple fit; second, whether it affords unobstructed spinning so that turns can be made. The two tests rely on three agent-relative movement constraints: the geometries of fit in the *x*- and *y*-direction, and spinning, with the latter implying the other two. These geometrical constraints depend both on body scales as well as any additional equipment necessary for movement, such as a wheelchair or walking stick. They have been extensively studied and have entered national guidelines for the design of indoor spaces: German DIN 18024-1 standard (http://nullbarriere.de/din18024-1.htm) provides useful quantifications (Fig. 1).

If both conditions hold, we say that the location affords full possibility of movement; if only the former is true, the location is a potential point on a path but affords no turns—movement can only proceed straight ahead in the *x*- or *y*-direction (provided, of course, that the adjacent location itself affords fit). What we obtain in this way is a network of locations with turn restrictions, suitable for routing; we term the set of all such locations *occupiable space* and the corresponding network *accessibility graph*.

The effect of running the tests at each location is similar to running vast numbers of agent-based simulations between pairs of locations, requiring however less time, generating no noise and, most importantly, providing exhaustiveness. It models the possibility of movement in the simulated space without reference to any individual paths: unlike agent-based simulations, mobility affordances do not emerge here as a result of executing (virtual) path taking but as a possibility for it. Furthermore, continuous testing across space is in accord with the conclusion that in indoor environments mobility affordances are better thought of as attributes of the continuous space itself rather than any one environmental object.

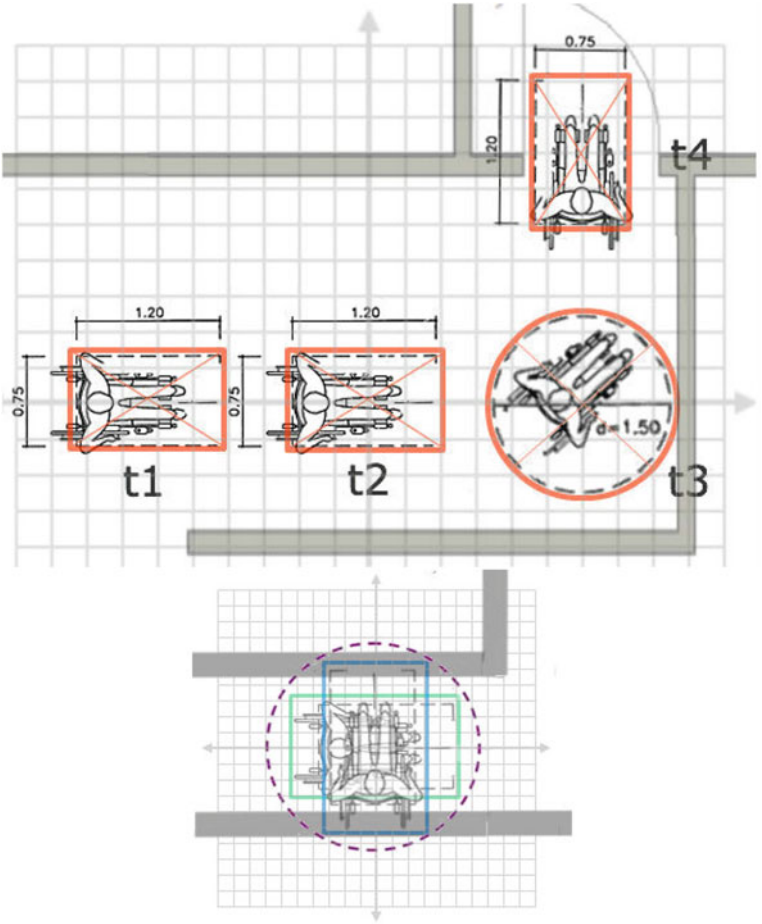


Fig. 1 *Top* Path taking can be thought of as a collection of discrete moments t_n. *Bottom* Movement primitives geometry: *x*-direction fit (*green*), *y*-direction fit (*blue*) and spinning (*violet*; based on DIN 18024-1); within this layout of obstacles (*grey*), only *x*-direction fit is afforded

3.2 Modelling the Environment

Referring to the discussion in Sect. 2, there are conflicting demands on our input spatial model: it should preserve the continuity of the indoor geometry yet either incorporate or enable the derivation of network-based descriptions of the environment and the running of network-based analyses. The various required conceptions of space as inputs and outputs for our analysis are shown in Fig. 2.

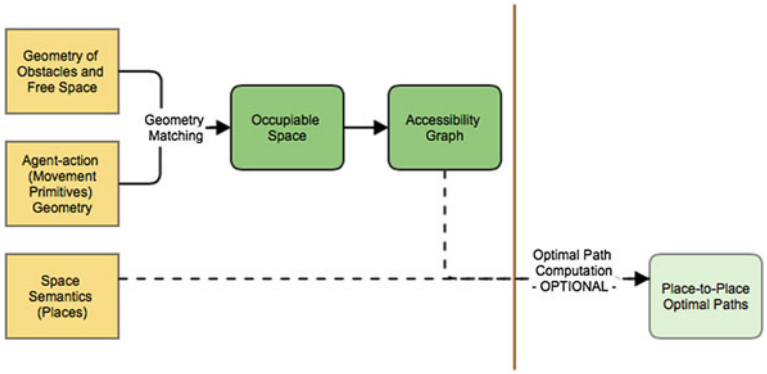


Fig. 2 Spaces as input (*yellow*) and output (*green*) of accessibility analysis

Our search for an adequate spatial model is guided by the comparative review presented in Afyouni et al. (2012), as well as the procedure outlined in Sect. 3.1. The geometry of obstacles can be captured with precision in a CAD model; however, its lack of the explicit encoding of empty space prevents easy action-environment geometry matching necessary to compute occupiable space. Moreover, standard network-based analyses such as optimal paths computation cannot be run on CAD models. The issue of continuous coverage of the indoor space can be resolved by using cell-based models (tessellations); additionally, these implicitly model spatial adjacency. This is crucial in our case: to establish the possibility of movement between locations by only testing the locations themselves for mobility affordances, we must ensure that locations—cells—adjacent in the model represent locations adjacent in reality.

We decided that a regular tessellation (grid) lends itself best to our method. While irregular cells in general capture the geometry of obstacles with more precision (as cell borders can trace non-orthogonal shapes), they also have an important drawback from the perspective of our goal. To test locations (cells) for movement affordances as described above, at each cell different neighbourhoods of $i * j$ surrounding cells model the agent-relative geometries of fitting in space and spinning. A straightforward automation of the testing procedure asks for a uniform cell size across the modelled space so that these action geometries can be consistently compared against the environmental geometry. One drawback is that non-orthogonal elements can only be approximated; the finer the granularity, the better the approximation.

To allow routing on the derived occupiable space, we propose to use the idea of the grid graph as outlined in Li et al. (2010). The grid graph model starts from a regular tessellation of an environment and then builds a base graph on top of it by treating each cell as a network node and the connections between it and the adjacent cells as edges to which weights are attached. In this way it offers the analytical advantages of network representations while retaining the continuity of the environmental geometry. Another welcome feature of the grid graph model is that it

stores the semantics of the environment by labelling each node with membership in a named architectural unit (place); the structure of places is modeller-specified. In this way place-based graphs of various levels of abstraction can be derived from the base graph.

Modelling the geometry of obstacles to movement requires a working definition of obstacle. These usually refer to architectural barriers—walls and ceilings—but an advantage of a continuous spatial representation is that fixtures and furniture too can be explicitly modelled. In this way a very precise computation of accessible space is possible: taking as an example our use case of a public library, the stack areas can be tested and fine routing performed on the resulting occupiable space.

Our procedure as outlined below is constrained to two-dimensional space. To take into account differences in the height coordinates of cells and the restrictions to movement in a wheelchair that this can cause, stairs and ramps were treated as obstacles from the outset, with the cells belonging to those areas excluded from the occupiable space computation.

To settle the issue of optimal grid resolution, we performed a comparative analysis of three different resolutions in our use case (Sect. 4) and compared them to the CAD source to determine how the difference affects the resulting occupiable space. This calibration of grid resolution would have to be performed for each environment being analysed.

3.3 Accessibility Assessment: Procedure

The outlined procedure was implemented in the TerraME modelling environment (Carneiro et al. 2011), with the conversion from CAD to tessellation done via the TerraView GIS. TerraME's CellularSpace class implements the grid graph idea by allowing the explicit modelling of connection weights between adjacent grid cells so that network analyses can be run on it. The algorithm runs in linear time O(n) for all outlined steps, with n being the total number of grid cells (Table 1).

Step 0. Convert the CAD files into a grid with granularity *g*; *g* should be divisible without remainder into the dimensions of agent-relative movement geometries (fit and spin), so that the geometries can be represented by whole numbers of cells. Unnecessary information should be manually removed from the CAD source so that all that remains are obstacles represented as polygons; it is then converted to a shapefile and finally to a tessellated representation (we used TerraView GIS for the final step as it offers a straightforward conversion procedure).
Each cell belongs either to empty space or an obstacle; this membership value is stored in the cell's *state* attribute. If a cell contains both obstacles

Table 1 Symbols used in pseudocode

g	Grid granularity
Q	Set of all grid cells
createNeighbourhood (*arg1, arg2*)	TerraME function assigning to each cell in a set (arg1) an array of pointers to cells based on a neighbour selection strategy (arg2); here, a cell is selected as neighbour if it is one of the $(i/g) * (j/g)$ surrounding cells
$i * j$	Dimensions of an agent-relative movement geometry
O_x, O_y, O_s	Occupiable spaces (sets of occupiable cells) for each movement geometry
$state[c]$	Cell c's membership in obstacle/empty space
$N_x[c]$, $N_y[c]$, $N_s[c]$	Cell c's neighbourhood arrays modelling movement geometries; results of *createNeighborhood*()
$occupiability_x[c]$, $occupiability_y[c]$, $occupiability_s[c]$	Attribute describing cell c's occupiability: one *occupiability* attribute for each movement geometry
$x[c]$, $y[c]$	Cell c's x- and y-coordinate in the grid
O	Total occupiable space (union of O_x, O_y and O_s)
Von Neumann n′hood	The four cells orthogonally adjacent to a cell in a 2D grid
$weight[c, n]$	Weight of an edge (connection between cell c and a Von Neumann neighbour n)

and empty space its membership value is decided based on their ratio—if non-empty space comprises 50 % or more of the cell's area it is counted as part of obstacle space.

Step 1. For each agent-relative movement geometry (x- and y-direction fit and spinning), assign the cells modelling it to each grid cell by creating a neighbourhood around the cell, for testing in step 2.

STEP 1

function ASSIGNMOVEMENTGEOMETRIESTOCELLS(Q) ▷ for each cell, create three neighbourhoods

▷ modelling the three movement geometries

$createNeighborhood(Q, (i_x * j_x)/g^2)$ ▷ x-fit: $i_x = 120$ cm, $j_x = 75$ cm

$createNeighborhood(Q, (i_y * j_y)/g^2)$ ▷ y-fit: $i_y = 75$ cm, $j_y = 120$cm

$createNeighborhood(Q, (i_s * j_s)/g^2)$ ▷ spinning: $i_s = 150$cm, $j_y = 150$ cm

Step 2. For each agent-relative movement geometry, compute the respective occupiable space. To do this, test at each cell whether all the assigned cells modelling the movement geometry belong to empty space; if so, the cell is labelled occupiable and added to the occupiable space. The pseudocode below shows the computation for *x*-direction fit; the procedure is identical for the other two movement geometries.

```
STEP 2
  function getOccupiableSpaceXDir(Q)
    initialise O_x as empty set
    for each cell c in Q do
      if state[c] = EMPTY then
        for each cell n ∈ N_x[c] do
          if state[n] = EMPTY then
            count ← count + 1
        if count = size(N_x[c]) then                    ▷ size(N[c]) = (i * j)/g^2
          occupiability_x[c] ← OCCUPIABLE
          add c to O_x
    return O_x
```

Step 3. Construct the accessibility graph on the total occupiable space by instantiating connections (edges) between occupiable cells (nodes). Each occupiable cell is connected to those of its orthogonally adjacent cells (i.e. cells comprising the cell's Von Neumann neighbourhood) towards which movement is afforded. This is established depending on the types of occupiability of both current and adjacent cell as well as the adjacent cell's location relative to the current cell. For example, based on the limitations of moving in a wheelchair, if a cell affords fit in the *x*-direction only (no spinning afforded), movement can only proceed in a straight path—that is, to an adjacent cell with the same *y*-coordinate, and only so if it too allows (at least) fit in the *x*-direction. Since movement was modelled for the *x*- and *y*-directions only, diagonally adjacent cells are not considered when constructing the graph; the encoded turns are therefore 90°. Each possible connection is assigned a weight equalling grid granularity *g*; impossible connections carry very high weights to avoid routing through them.

STEP 3

function CONSTRUCTACCESSIBILITYGRAPH(O)
  for each cell c in O **do**
    if $occupiability_{\mathrm{s}}[c] = OCCUPIABLE$ **then** ▷ spinning at c possible
      for each Von Neumann neighbour n of c **do**
        if ($x[n] = x[c]$ and $occupiability_{\mathrm{y}}[n] = OCCUPIABLE$) or ($y[n] = y[c]$ and $occupiability_{\mathrm{x}}[n] = OCCUPIABLE$) **then**
          $weight[c, n] \leftarrow g$ ▷ weight is equal to grid granularity
        else
          $weight[c, n] \leftarrow infinity$ ▷ any very large value
    else if $occupiability_{\mathrm{x}}[c] = OCCUPIABLE$ **then** ▷ x-direction fit at c possible
      for each Von Neumann neighbour n of c **do**
        if $y[n] = y[c]$ and $occupiability_{\mathrm{x}}[n] = OCCUPIABLE$ **then**
          $weight[c, n] \leftarrow g$
        else
          $weight[c, n] \leftarrow infinity$
    else if $occupiability_{\mathrm{y}}[c] = OCCUPIABLE$ **then** ▷ y-direction fit at c possible
      for each Von Neumann neighbour n of c **do**
        if $x[n] = x[c]$ and $occupiability_{\mathrm{y}}[n] = OCCUPIABLE$ **then**
          $weight[c, n] \leftarrow g$
        else
          $weight[c, n] \leftarrow infinity$

4 Case Study: ULB

As a test case, we dealt with the university and state library (Universitäts- und Landesbibliothek, ULB) in Münster. In this chapter we shortly discuss the results.

A preliminary occupiable space computation for a section of the ULB was performed on grids with granularity 5, 7.5 and 10 cm respectively (Fig. 3). A comparison to the CAD source revealed that while there was virtually no difference between the former two in correctly identifying overall accessibility, the

Fig. 3 Occupiable space depending on grid resolution (**yellow**): **a** 5 cm, **b** 7.5 cm, **c** 10 cm; *red circles* mark existing connections lost due to lower granularity

latter rendered some accessible door spaces inaccessible: the coarser resolution meant that some empty space was lost in CAD-grid conversion by ending up in cells mostly comprised of obstacle space (since the conversion used percentage of total area as the criterion in assigning each cell to empty vs. obstacle space). Only a few centimetres lost, however, meant that the constraints of action (passing through) were no longer satisfied. In order to balance geometrical precision with performance (significantly fewer cell count), a resolution of 7.5 cm was chosen for the subsequent analysis.

As a general rule of thumb one should consider that the standard minimum door width for unobstructed wheelchair movement according to the norm (DIN 18024-1 standard, http://nullbarriere.de/din18024-1.htm) is 80 cm, and that it is particularly important to make sure the space of doors of this width is identified in the computation as allowing movement. As explained in step 0 above, in the process of conversion of a CAD source the overlay of the source with a grid of cells inevitably results in some loss of otherwise empty space in cases where a cell covers an area consisting of both empty and non-empty space. In the worst-case scenario a door width of 80 cm can be overlaid in such a way that both doorjambs end up in cells assigned to obstacle space. With a 10 cm-grid up to 10 cm of empty door space can be lost (5 cm on each side) and the door would subsequently not be identified as a valid link; with a 7.5 cm-grid the largest possible loss is only 5 cm (3.75 and 1. 25 cm on both sides in the worst case, respectively), which still makes it possible for our algorithm to capture it as a valid connection.

Figure 4 shows the different ways in which locations (cells) can afford mobility. An agent can simply fit into the surrounding space (in the *x*- or *y*-direction) or the full possibility of turning can be afforded. All three subsets of the overall space are needed to model movement and routing algorithms have to take into account the resulting turn restrictions. Figure 5 shows a shortest path computed using a Dijkstra algorithm. Turns can only occur at those cells that have come out from the occupiable space computation as allowing turns.

Fig. 4 Occupiable space (*yellow*): **a** x-direction only, **b** y-direction only, **c** full turn possible

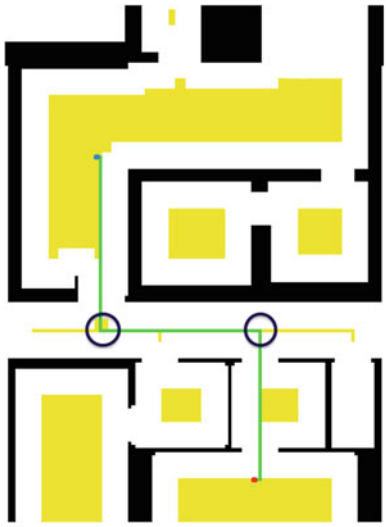


Fig. 5 Shortest path (*green*) between start (*red*) and goal (*blue*) cells with turn restrictions: the two necessary turns (*circled*) are constrained to cells affording full spinning (*yellow*)

5 Discussion

5.1 Limitations of the Method and Possible Improvements

To fully assess the accessibility of an indoor environment an extension into the third dimension is needed. The present approach assesses vertical connectivity by testing whether elevators afford wheelchair access; cells belonging to the same elevators in different floors can thus be considered connected. In the vertical dimension, the movement of wheelchair users can however be impeded by features such as low ceilings, or even door handles or elevator buttons that are out of reach. A 3D extension of our approach would make it redundant to label stairs and ramps as obstacles in advance and exclude them from the occupiable space computation: ramps could be assessed based on the slope and stairs on the riser/tread ratio within the procedure itself, meaning less effort in the preparation step. As a first extension of our approach, cross-sections of buildings in CAD format can be used and movement primitives (geometries of fit) in the vertical dimension modelled.

A refinement of the method to compute occupiable space is possible. For our present purposes, a rather crude division between fit in the x- and y-direction and full spinning was used. One way to improve this would be to break down full spinning occupiability into four turning possibilities (90° turns) and test locations for those separately. Such partial turn restrictions in the accessibility graph would allow finer routing. Additionally, non-9° turns can be taken into account by considering the diagonal neighbours of cells, and for this purpose non-90 fit geometries would be used.

5.2 Integration of Procedure Results into an Indoor Navigation System

The outlined procedure is only a first step towards realising a full-fledged navigation aid and the information obtained through it can be used in different ways depending on the way space is encoded in the navigation system in question.

Through steps 1–3 of the procedure we can obtain the set of occupiable nodes and a routing graph for wheelchair users. Depending on the indoor environment a large number of inaccessible nodes can be removed from subsequent computations thus relieving some of the computational costs associated with large numbers of nodes that the fine resolution of the graph implies. Routing can then be performed in a number of ways. The simplest one would be to use a variant of the Dijkstra algorithm that implements turn restrictions, as was done above. Wheelchair users can benefit from least effort paths in addition to shortest, so additional costs for turns can be encoded. Moreover, a less greedy search such as the A* algorithm with the Euclidean distance heuristic can be used to further increase computational performance.

On the other hand, if issues of memory usage are paramount and there is no space for a complex and memory-intensive model such as the grid graph, we can extend the outlined procedure to come up with a pre-computed set of shortest paths accessible for a wheelchair user; this would then constitute the system's sole spatial knowledge base. We begin by using the semantic information on the indoor environment encoded at cell level: each cell is a member of a place (e.g. room or corridor; see Sect. 3.2). For each pair of places we run a shortest path computation, with a randomly chosen occupiable cell within each place as start/goal cell. The resulting path geometries are then turned into semantic path descriptions by querying the constituent cells for their membership values. Routing can then be done simply by retrieving the path description for each start-goal place input. Moreover, the path geometries encode metric information that can be used to compute the time cost of each path as another piece of semantic information.

Referring to Figs. 3 and 4, we can see that although some places are part of the overall occupiable space in virtue of allowing wheelchair users to move within them, they are cut off from the rest of the occupiable space when their doors do not allow wheelchair passage. The shortest path computation as described above is able to identify such cases, and the user can be notified in advance of the places that cannot be accessed.

If the navigational system uses a model based on semantics (see Afyouni et al. 2012 for a thorough review of those) such as a place graph, the results of our procedure can be entered as annotation. Places can be tagged for accessibility in a similar way outdoor elements are in WheelMap (http://wheelmap.org/), while their connections can be labelled with optimal distances and times resulting from a computation such as the one outlined above. To achieve this, however, we first need a definition of what makes a place such as a room accessible. A working definition could be that belonging to the occupiable space of the building and being connected

to at least one more place via the occupiable space justifies the 'accessible' tag, but this remains open for discussion. It is also possible to include occupiable space as a category in indoor space ontologies to allow reasoning on navigation-related questions as outlined in Höllerer et al. (2001). Since affordances are fundamentally about meaning, ontologies are the right places for the results of their automated derivation to come to full fruition (Ortmann and Kuhn 2010).

6 Conclusion

Indoor navigation systems can be of great help to the disabled, provided they adopt the distinctive perspective on navigation of these special user groups. Following the procedure outlined in this paper, we have been able to ascertain the extent to which an indoor environment is accessible to wheelchair users as well as lay the foundations for a routing system that takes into account the particularities of movement of this section of the general population. The approach integrates two conceptual models of space: continuous spatial representation required for movement affordance computation, and place-based view used in everyday navigation. It can be used as a general method for affordance computation based on action geometries as parameters and run on a regular grid representation of the space in question. Its most important outcome is its user-relative (adaptive) nature that can be used as the basis for mobile assistance systems for different locomotion types. Future work will concentrate on improving the accessible space computation by refining the movement primitives to align them more to the way wheelchair users actually move. We plan to use the results of our case in developing a resource navigator application for the ULB.

Acknowledgments The work presented in this paper was conducted and financed as part of the LIFE project at the Institute for Geoinformatics (IFGI) of the University of Münster. The Universitäts- und Landesbibliothek were kind enough to provide floor plans for the library building. The authors owe gratitude to Dr. Pedro Ribeiro de Andrade of Brazil's National Institute for Space (INPE) for his help with programming in the TerraME modelling environment, as well as Dr. Marco Painho of the NOVA School of Statistics and Information Management (ISEGI-NOVA) and Dr. Rui Li of IFGI for their valuable input.

References

Afyouni, I., Ray, C., & Claramunt, C. (2010) A fine-grained context-dependent model for indoor spaces. In *Proceedings of 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*.

Afyouni, I., Ray, C., & Claramunt, C. (2012). Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science, 4*, 85–123.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577–660.

Barsalou, L. W. (2003). Situated simulation in the human conceptual system. *Language and Cognitive Processes, 18*(5/6), 513–562.

Budziszewski, P., Grabowski, A., Milanowicz, M., Jankowski, J., & Dzwiarek, M. (2011). Designing a workplace for workers with motion disability with computer simulation and virtual reality techniques. *International Journal on Disability and Human Development, 10*(4), 355–358.

Carneiro, T., Camara, G., de Andrade, P. R., & Pereira, R. R. (2011, February). An introduction to terrame. In *INPE and UFOP Report, Version 1.5*.

Chemero, A. (2003). An outline of a theory of affordances. *Ecological Psychology, 15*(2), 181–195.

Fallah, N., Apostolopoulos, I., Bekris, K., & Folmer, E. (2013). Indoor human navigation systems: A survey. *Interacting with Computers, 25*(1), 21–33.

Gibson, J. J. (1977). The theory of affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving* (pp. 67–82)., Acting, and knowing: Toward an ecological psychology Hillsdale, NJ: Lawrence Erlbaum.

Höllerer, T., Hallaway, D., Tinna, N., & Feiner, S. (2001). Steps toward accommodating variable position tracking accuracy in a mobile augmented reality system. In *AIMS'01: Second Int. Workshop on Artificial Intelligence in Mobile Systems* (pp. 31–37), Seattle, WA, August 2001.

Jonietz, D., Schuster, W., & Timpf, S. (2013) Modelling the suitability of urban networks for pedestrians: An affordance-based framework. In D. Vandenbroucke et al. (Eds.). *Geographic information science at the heart of Europe. Lecture Notes in Geoinformation and Cartography*. Switzerland: Springer.

Jonietz, D., & Timpf, S. (2013) An affordance-based simulation framework for assessing spatial suitability. In T. Tenbrink et al. (Eds.), *COSIT 2013. LNCS 8116* (pp. 169–184). Switzerland: Springer International Publishing.

Li, X., Claramunt, C., & Ray, C. (2010). A grid graph-based model for the analysis of 2d indoor spaces. *Computers, Environment and Urban Systems, 34*, 532–540.

Neufert, E. (2005). *Bauentwurfslehre*. Wiesbaden: Vieweg Verlag.

Ortmann, J., & Kuhn, W. (2010). Affordances as qualities. In *Proceedings of the 2010 conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010)*.

Raubal, M. (2001a). Agent-based simulation of human wayfinding: A perceptual model for unfamiliar buildings (Ph.D. Thesis, Vienna University of Technology).

Raubal, M. (2001b). Ontology and epistemology for agent-based wayfinding simulation. *International Journal of Geographical Information Science*, *15*(7).

Raubal, M. (2008). Wayfinding: Affordances and agent simulation. In *Encyclopedia of GIS*.

Raubal, M., & Worboys, M. (1999). A formal model of the process of wayfinding in built environments. In C. Freksa, D. M. Marks (Eds.), *COSIT'99. LNCS 1661* (pp. 381–399).

Richter, K.-F., Winter, S., & Santosa, S. (2011). Hierarchical representations of indoor spaces. *Environment and Planning B: Planning and Design, 38*(6), 1052–1070.

Rüetschi, U.-J., & Timpf, S. (2005). Modelling wayfinding in public transport: Network space and scene space. In C. Freksa et al. (Eds.), *Spatial Cognition IV. LNAI 3343* (pp. 24–41). Heidelberg: Springer.

Scarantino, A. (2003). Affordances explained. *Philosophy of Science, 70*, 949–961.

Scheider, S. (2011). Grounding geographic information in perceptual operations (Ph.D. Thesis, Westfälische Wilhelms-Universität Münster).

Scheider, S., & Janowicz, K. (2014). Place reference systems. *Applied Ontology, 9*, 97–127.

Swobodzinski, M., & Raubal, M. (2008). An indoor routing algorithm for the blind: Development and comparison to a routing algorithm for the sighted. *International Journal of Geographical Information Science, 00*(00), 1–28.

Warren, W. H. (1984). Perceiving affordances: Visual guidance of stair climbing. *Journal of Experimental Psychology*, *10*(5).

"Turn Left" Versus "Walk Towards the Café": When Relative Directions Work Better Than Landmarks

Jana Götze and Johan Boye

Abstract An automatic mechanism that gives verbal navigation instructions to pedestrians in situ needs to take into account a number of factors. Besides giving the instruction at the right time and place, the information needs to be as unambiguous as possible for the user to both choose the correct path and be confident in doing so. Humans make extensive use of landmarks when describing the way to others and are more successful following instructions that include landmarks. We present a study comparing landmark-based instructions with relative direction instructions on pedestrians in a real city environment, measuring both objective and subjective success. We find that at some decision points, relative direction instructions work better. We present a method that uses openly available geographic data to predict which kind of instruction is preferable at a given decision point.

Keywords Route instructions · Wayfinding · Pedestrians · Landmarks · Open geographic data

1 Introduction

Giving and following route instructions are tasks that most of us carry out regularly. Nowadays, many people have a smartphone with a built-in GPS receiver. This new state of affairs, along with the increasing coverage and quality of open geographic databases [like OpenStreetMap (Haklay and Weber 2008)], has opened up the possibility of implementing systems that give real-time verbal routing instructions for the city pedestrian. When there is no map at hand, receiving well-timed instructions at the scene can present an improvement compared to, say, having to

J. Götze (✉) · J. Boye
School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: jagoetze@kth.se

J. Boye
e-mail: jboye@kth.se

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_15

look at printed instructions on a sheet of paper, or having to memorize the entire route beforehand. An advanced verbal routing system might also be interactive and able to understand requests for clarification from the user (e.g. Boye et al. 2014; Janarthanam et al. 2012). Such systems are in many cases able to provide an alternative explanation in case the user has not understood where to go next.

To minimize the number of misunderstood instructions and ensuing clarification requests, it is of great importance for system designers to come up with algorithms which will generate the most successful instruction in any given situation. By 'successful' we mean both objectively successful (in the sense of leading to as few navigational errors as possible) and subjectively successful (in the sense of user satisfaction and perceived confidence). Furthermore, instructions should be easy for the pedestrian to process. They should be interpretable without a map and, as we are aiming for practical applications, possible to generate from freely available geographic databases.

In this paper, we present a study carried out to provide some empirical basis for such an instruction-giving algorithm. Our aim is to compare the effectiveness of landmark-based instructions with relative direction (left/right/straight) instructions (and with combinations such as "Turn left towards the X"). Several studies have pointed out that landmarks play a special role in the communication of route directions (Allen 1997; Caduff and Timpf 2005; Denis 1997; Michon and Denis 2001; Raubal and Winter 2002; Richter 2008; Tom and Denis 2004). However, direction givers do not always prefer to give a landmark-based instruction and by studying in situ route instructions given by people, we noted that this seems to depend on where the instruction is given. We are now interested in how such relative directions are interpreted by someone who has to follow them, and under which circumstances such instructions work well.

Furthermore, as a practical consideration, the most preferable (or salient) landmark might not be in the database (e.g. the billboard with the big clock is not currently (Nov. 2014) represented in OpenStreetMap (see Fig. 1). Another

Fig. 1 The billboard with the clock is a salient landmark not present in the geographic database

possibility is that a salient landmark (e.g. a building) is present in the database, but the feature that distinguishes it (e.g. its red color) is not represented. Thus, there are reasons to explore other kinds of instructions than landmark-based ones.

2 Background

2.1 Route Directions

How people give and understand route directions has been studied extensively. While there is no single definition for good route directions, several aspects have been found to play a role:

A route is typically split into several segments that are then verbalized (Couclelis and Portugali 1996). These verbalized directions can be instructions to take a particular action, such as "walk" or "turn", or descriptions of the environment like "There is a red building to your left." that help the direction follower to identify where an action is to be carried out or whether he is still on the right way (Allen 1997; Denis 1997). The order of the directions should reflect the linear order in which the route is traversed (Allen 2000). These conclusions have been drawn from verbal or written route directions that have been produced prior to someone following them, i.e. the route follower had to memorize the directions before carrying them out.

Aspects of route direction following have also been studied in a setting where directions are given while the follower proceeds along the route. Hund and Minarik (2006) as well as May and Ross (2006) have studied route directions for car navigation, and several systems have been built for direction giving in a virtual environment (cf. Striegnitz et al. 2011), focusing mostly on the generation of referring expressions. For directions given to pedestrians, studies typically involve the direction follower to find a route on a map or draw a route on paper (Li et al. 2014; Michon and Denis 2001; Tom and Denis 2004).

2.2 The Role of Landmarks in Route Directions

That landmarks play a special role in the communication of route directions has been demonstrated in several studies. Landmarks are a means to identify crucial points along the route where turning actions need to be taken or could be taken (decision points), as well as to locate the beginning and the end of the route (Michon and Denis 2001). Landmarks also play a role in the descriptive part of route directions, to locate other landmarks, and to confirm that the follower has correctly executed a turn (Daniel and Denis 1998). Salient landmarks can also be used as a criterion to find the best route in terms of how easy it will be to follow (Caduff and Timpf 2005; Richter 2008).

The term landmark is usually used for a visually salient object along the route that is easily identifiable by some outstanding feature, such as its size, color, or special function. While street names do not serve as good landmarks because they cannot easily be identified (Tom and Denis 2004), streets themselves can serve as landmarks if they are described vividly in terms of their outstanding features (Tom and Tversky 2012).

Raubal and Winter (2002) try to compute the salience of landmarks using different salience measures to reflect visual, structural, and semantic salience. However, the information they use is often not readily available, such as a landmark's color, façade, or cultural importance. We (Götze and Boye 2013) have used information that is freely available from OpenStreetMap, such as a landmark's position and type, to compute its salience.

2.3 Comparing Different Types of Instructions

While it is undisputed that landmarks play a vital role in making route directions easy to follow, automatically choosing an appropriate landmark in a given situation is not an easy task. There is often insufficient information available to decide which landmark stands out in terms of salience, such as its color, height, or special function.

Looking at human route directions, we find that at some decision points, the describers seem to prefer to not use a landmark. In Daniel and Denis' study (1998), about 14–20 % of route directions did not include a landmark. In Rehrl's study (Rehrl et al. 2009), in which participants were giving directions in situ, while walking along the route, only 1–7.4 % of the directions were instructions without a landmark. However, this reflects only the preference of the direction giver to include landmarks. We are interested in knowing how useful such an instruction is for a direction follower.

3 Modeling the Spatial Environment

Let us consider the natural hypothesis that if the decision point has a simple configuration, such as a T-intersection or a four-way intersection where all streets meet at right angles, an instruction containing a relative direction will be sufficient. By contrast, if the decision point has a more complex configuration, such as many streets meeting at non-right angles, the inclusion of a landmark in the instruction will yield higher confidence and fewer errors. For instance, in the situation depicted in Fig. 2a, the instruction "please turn left" would be sufficient, whereas in Fig. 2b it would not.

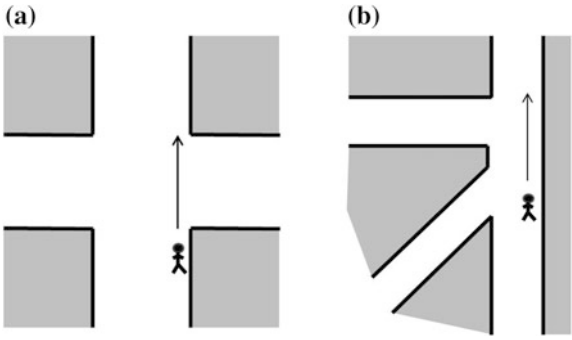


Fig. 2 **a** (*left*) A situation where "turn left" is probably clearly comprehensible, and **b** (*right*) where it is not. The *arrows* indicate the walking direction

Though it is easy to draw such idealized maps and make conjectures about which instructions would be suitable in the clear-cut cases, it is less obvious how to devise a general algorithm for deciding, for any given situation, whether an instruction should be couched as a relative instruction or a landmark-based instruction. Moreover, such an algorithm must work in real time, and only use the information present in open geographic databases like OpenStreetMap.

OpenStreetMap has (basically) two data types: *nodes* and *ways*. Ways are sequences of nodes, used for representing a wide variety of objects, such as roads, squares, areas and buildings (in the three latter cases, the first node in the sequence is the same as the last node, and hence the way forms the perimeter of a polygon). An intersection between two streets is represented by the point where the ways corresponding to the streets meet. However, the situation in Fig. 2b (and similar situations such as roundabouts), that for the human eye constitutes a single complex intersection, have no explicit representation in OpenStreetMap. This has to be taken into account when designing an algorithm for instruction generation from geographical databases.

Our system for pedestrian routing instructions (Götze and Boye 2013) contains a visibility engine, which can quickly perform line-of-sight computations. We use this engine to compute, for a given situation, **how far the closest building is** in every direction from −100° to 100°, relative to the user's current direction (cf. Fig. 3). This vector of 201 numbers thus constitutes a crude representation of the user's spatial surroundings. One of our aims is now to investigate whether there is a systematic relationship, on the one hand between the numbers of this vector for a given situation, and on the other hand how test subjects perceive a relative instruction in that situation.

In order to reduce the dimensionality of the input data, we divide the user's field of vision into 7 subsectors: straight ahead (–10° to 10°), as well as 11°–40°, 41°–70°, and 71°–100° on either side, and only consider the maximal distance in each subsector. For example, a typical such vector could be (252, 20, 10, 200, 14, 2, 1) from left to right, meaning that the user has a free line of sight extending 252 m to the left in the sector −100° to −71°, 20 m in the sector −70° to −41°, and so on.

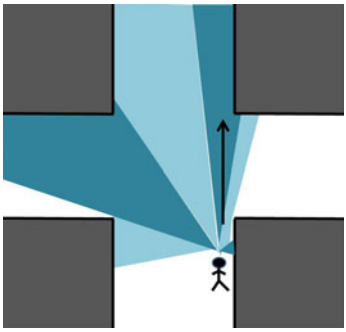


Fig. 3 The considered sectors of the user's field of vision

4 Experiment

In order to gain an insight into where it is preferable to use a landmark in an instruction, we asked a number of subjects to follow different kinds of route instructions. After each instruction they received, they rated their confidence in knowing which direction to choose. The instructions they received consisted of either a relative direction ("Turn left/right/straight"), or a landmark ("Walk towards the school"), or they combined both pieces of information.

Materials
We have determined two routes with slightly different street layout for this study. Route I contains 14 decision points with more complex configurations, such as a roundabout or several streets turning into similar directions. Route II contains 13 decision points with simpler configurations, where streets meet at right angles.

We constructed two sets of instructions for both routes. The first set contained only relative instructions of the form "Turn/Go <direction>". We call this set the *direction set D*. The second set, the *landmark set L*, contained instructions that used only a landmark, e.g. "Walk towards the school". For Route I, we also constructed a third set that combined both pieces of information into one, we call this set the *landmark+direction set LD*. The instructions for each route were then randomly arranged into two versions (Route II) of route instructions (three versions for Route I) with the only restriction that there should not be more than three instructions of the same kind following each other. Table 1 shows example instructions for each of the three sets of instructions.

In this study, we are focussing on only one part of route directions, the action prescriptions (cf. Daniel and Denis 1998). For route directions to be complete, they need to also contain a description of how to identify the decision point where an action is to take place. Here, we are only interested in the use of landmarks to carry out the action at a decision point. In the navigation system described below, we have access to the pedestrian's position through the GPS signal and can take part of the burden to locate the decision point off the pedestrian by saying "walk until I say stop".

Table 1 Example instructions for each of the three instruction types

Type	Example Instructions
Direction	Go straight Turn right
Landmark	Go towards the school Go down the stairs Walk along the big street
Direction + landmark	Go straight towards the school Turn left towards the bicycles Go down the stairs on the right

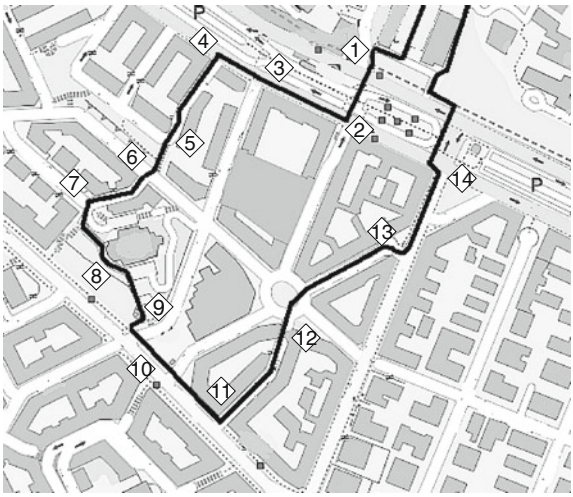


Fig. 4 A map of Route I, containing 14 decision points. *Map source* © OpenStreetMap contributors

Route I can be seen in Fig. 4. This route, as well as the landmarks for the two landmark-based conditions, comes from one of our previous experiments in which participants were asked to describe the route while they were walking along it (Albore et al. 2013). Route I was about 1.1 km long and contained 4 left turns, 3 right turns and 7 continuations, i.e. decision points where no change of direction occurred. Decision points were included at each point where a change of direction could occur. Most of the continuations were explicitly mentioned by the participants of our previous experiment and therefore included as decision points. This segmentation also conforms to the way that an automatic system could choose decision points. In Götze and Boye (2013) all nodes that initially belong to the route are chunked based on visibility and a limit on distance between them. If the angle between two segments is very small, the direction will be "straight".

Route II contains 5 left turns, 6 right turns, and 2 continuations. For this route, continuations were omitted as explicit instructions, instead, the participant received a confirmation of the form "You are doing fine. Please keep walking straight". This confirmation was not rated by the participants. The landmarks are selected manually and vary between types and names of shops, and descriptions of streets.

The participants were equipped with a Samsung Galaxy S4 mobile phone running an application that sent the participant's position and speech data and received text that was synthesized by the phone's Text-To-Speech application (TTS). The experimenter took the role of a wizard in a so-called Wizard-of-Oz (WoZ) data collection (Dahlbäck and Jönsson 1989), and interacted with the participant by selecting what to say from a pre-determined set of utterances. When the participant reached a decision point, the experimenter selected the corresponding button that started a small dialog with the participant, asking him to stop, listen to the instruction, and rate it. An additional window allowed the experimenter to monitor the participant's position trail as well as possibly problematic situations such as interruptions in the connection between the phone application and the WoZ interface.

Participants

18 participants (6 female and 12 male, average age = 29, SD = 3.3) followed Route I in return for a cinema ticket. 14 participants (6 female, 8 male, average age = 24.9, SD = 2.3) followed Route II in return for course credit. For both routes, participants were randomly assigned one of three (Route I) or two (Route II) different sets of route instructions. None of them had participated in any of our previous experiments. None of them followed both routes.

Procedure

Participants were asked to follow a set of instructions, each of which described the next direction to take. Each instruction was rated before they carried out the action it described.

The participants, after receiving an instruction, were asked to choose the direction that they thought was correct and walk straight until the system told them to stop to get the next instruction. In this way we could make sure that everyone got the instruction at the same point and that the spatial environment at the time of the instruction was stable and did not change due to the participant's movement.[1] Until the participant had rated the instruction, he or she was standing still at a decision point.

The process of the experiment was the following: Each participant was equipped with a mobile phone running an application that connected to software running on the experimenter's computer, as well as a headset. The participant was informed that she would receive a set of navigation instructions through the phone, which she would have to rate on a scale from 1 to 5, reflecting her confidence in knowing which action to perform. 1 corresponds to the lowest possible confidence ("not confident at all"), 5 to the highest ("very confident"). The participant was then asked to carry out the action and was welcomed to leave spoken comments. She was asked to keep walking straight until the system asked her to stop for the next instruction.

[1]Note that there is still some variability in the GPS signal.

Each participant was asked to step outside the building, where the experiment started with the first instruction. The first three or four instructions[2] were training instructions to accustom the participant to the synthetic voice and the order in which things were happening. They were not informed about the actual start of the experiment. At each decision point, the participant received detailed task instructions. Some of these were very detailed and as soon as the participant signalled that she was comfortable with the task, e.g. by barging in, she received a shorter version of the task description.

Collected Data
Of the 18 participants, 14 completed all 14 route segments. This results in a total of 232 confidence ratings. A rating is annotated with an error tag if the participant chose another than the intended next route segment. Furthermore, we have time-stamped log files that contain GPS data, as well as all instructions and other prompts (e.g. those that ask for a rating) played to the participant.

5 Experiment Results

The upper part of Table 2 shows an overview of the confidence ratings of Route I that the users gave for each of the 3 conditions, Landmark (L), Direction (D), and Landmark + Direction (LD), sorted by the kind of turn the follower had to take (*straight*: ↑, *left*: ←, and *right*: →). Average ratings for a segment in a given condition consist in 2 cases of 4 ratings, in 16 cases of 5 ratings, and in 24 cases of 6 ratings, and range from 2.17 to 5.0. Individual ratings range from 1 to 5. The average rating for all segments and conditions is 4.14, the median rating is 5.

The first seven rows show the continuation segments in which no change of direction occurred. In five of these cases, the D instruction was rated highest, with average ratings ranging from 4.2 to 5.0. In the other two continuation segments, the D instruction was rated 4.6 and 4.83, i.e. received a high rating as well. The overall average in ratings is 4.72. L instructions received an average of 3.85, this difference is significant ($t(61) = 3.53$, $p < 0.001$). The average rating for the LD instructions in these segments is 4.29. Instructions that included a landmark additionally to the relative direction were rated lower ($t(67) = 2.19$, $p < 0.04$). The difference in ratings between the L and the LD condition is not significant ($t(72) = 1.62$, $p = 0.11$).

In the segments where a change of direction occurred (cf. the middle of Table 2), the D instructions are rated highest only once, in segment 11 (4.33). The LD instructions receive higher ratings (4.44) than both the D instructions ($t(77) = 2.59$, $p < 0.02$) and L instructions ($t(68) = 2.19$, $p < 0.04$).

These two kinds of segments (direction change vs. no direction change) also differ in the number of errors that were made. Followers made more errors in the

[2]This varied due to some construction work that we needed to circumnavigate to get the participants to the starting point.

Table 2 Route I average ratings at each of the decision points for each of the three strategies

			Confidence ratings				
	Turning direction	Segment	Direction (D)	Landmark (L)	Landmark + Direction (LD)	Average[a]	# Errors
Route I	↑	1	**5.00**	**3.67**	**3.83**	4.12	1
	↑	3	**4.40**	**2.33**	**3.60**	3.38	1
	↑	5	**4.60**	**5.00**	**4.83**	4.82	
	↑	6	**4.83**	**4.50**	**5.00**	4.76	1
	↑	9	**4.20**	**2.17**	**3.33**	3.18	2
	↑	10	**5.00**	**4.67**	**4.80**	4.82	
	↑	14	**5.00**	**4.80**	**4.80**	4.86	1
	↑ Average		**4.72**	**3.85**	**4.29**	4.27	
	←	4	**4.60**	**4.83**	**5.00**	4.82	1
	←	7	**2.80**	**3.83**	**3.83**	3.53	3
	←	11	**4.33**	**4.20**	**4.00**	4.18	2
	←	13	**3.50**	**2.50**	**4.00**	3.27	3
	→	2	**4.67**	**2.40**	**4.83**	4.06	2
	→	8	**3.17**	**5.00**	**4.67**	4.24	2
	→	12	**3.00**	**4.00**	**4.67**	3.88	3
	←/→ Average		**3.74**	**3.82**	**4.44**	4.01	
		All	4.22	3.85	4.37	4.14	22
Route II	←	2	**4.71**	**4.71**		4.71	
	←	6	**5.00**	**3.86**		4.27	
	←	9	**5.00**	**4.00**		4.50	1
	←	12	**4.86**	**2.57**		3.71	1
	←	13	**5.00**	**4.86**		4.93	
	→	3	**5.00**	**3.86**		4.43	
	→	5	**4.40**	**3.86**		4.08	
	→	7	**5.00**	**3.60**		4.42	
	→	8	**4.86**	**3.71**		4.29	
	→	10	**5.00**	**4.71**		4.86	
	→	14	**4.86**	**3.29**		4.07	1
	←/→ Average (all)		4.89	3.92		4.40	3

D relative direction, *L* landmark, *LD* direction and landmark combined

[a]Recall that in some cases, the number of ratings for each strategy differs in the same segment

segments with a direction change, with 8 errors occurring after a D instruction, 6 occurring after an L instruction and 2 after an LD instruction. Errors in the segments without a direction change were never made in the D condition. The overall ratings for all segments differ between the LD instructions (4.37) and the L instruction (3.85; $t(142) = 2.73$, $p < 0.01$).

The lower part of Table 2 shows the results for Route II. We have collected 73 ratings for the D condition and 75 ratings for the L condition (14 participants have followed the instructions). Average ratings for a segment in a given condition consist in 2 cases of 5 ratings and in all other cases of 7 ratings. Average ratings range from 2.57 to 5.0, individual ratings range from 1 to 5. The overall average of

ratings is 4.40, the median rating is 5. The landmark-based instructions receive a confidence rating of 3.92 and the relative directions an average of 4.89 ($t(89) = 5.96$, $p < 0.0001$). Only three errors have been made, all after a landmark-based instruction.

The confidence ratings and errors that the participants following Route I made suggest that a relative direction is the best choice when going straight while followers are more confident with a landmark instruction if there is a change of direction. Overall, including both kinds of information works best both in terms of confidence ratings and number of errors.

The fact that a simple "straight" works best when no change of direction occurs is not surprising, especially for this particular route. Most decision points are complex intersections and going straight is both the simplest alternative to identify at many decision points and the default action, even without any instruction. An example for this can be found in segment 5, when the relative direction receives a lower average rating than the landmark-based instruction. The path that the participants are asked to follow consists of a flight of stairs that clearly lead away from the road they are walking along and that are easy to identify as "stairs" (hence the high L rating of 5.0). On the other hand, this is also the only possibility to continue straight and the D rating is not much lower (4.6).

However, this pattern may result from this particular route: most decision points are not simple four-way or T-intersections but more complex configurations, like a roundabout where six streets meet and thus a relative direction is ambiguous, even if it contains an additional modifier, such as "slightly right". Using a relative direction might work considerably better at a simple decision point, where streets meet at right angles. At the same time, some of the landmarks in the continuation segments may be inadequate and cause low confidence ratings because the follower cannot unambiguously identify them. We picked the landmarks from other peoples' descriptions, but did not account for whether they included the direction explicitly as well.

Route II contains decision points with simpler configurations where all streets meet at right angles. The ratings that we have collected suggest that for this route, the relative directions result in higher confidence than the landmark-based instructions.

6 Deciding What Instruction to Give

The confidence ratings we obtain from these studies suggest that both the street configuration at the decision point and the kind of action the pedestrian is supposed to carry out (turning left or right) influence their confidence in a certain type of instruction. A method that can predict the confidence score for a relative direction instruction from the geographical context and the turning direction will be a useful tool to generate the most helpful navigation instruction in a given situation. We show how we can employ linear regression to build such a model from the data we have collected.

The dependent (output) variable we want to be able to predict is the confidence score for a relative direction (D) instruction. As explained previously, this score is a number 1–5 representing how intelligible a relative instruction is perceived in the particular situation at hand. The independent (input) variables should reflect two aspects of the decision point: the street configuration from the follower's point of view, and the turning direction (we will only consider turns here and disregard the situations where the user is supposed to walk straight). We have shown in Sect. 3 how we can model the geographical context: Each decision point is represented by a vector of seven numbers, indicating the farthest distance to a building (or similar three-dimensional structure) in different directions from the user's current bearing. To include information about the turning direction into this vector, we are rearranging the vector as follows. The first three numbers represent the distances in the turning direction, the fourth number represents the distance straight ahead, and the last three numbers represent the distances on the side opposite of the turning direction. We thus have 7 independent variables $x_1 \ldots x_7$, representing distances in the various directions, and one dependent variable y, representing the estimated confidence score. In order not to over-fit the model, we use the integer value of the natural logarithm of the distances rather than the exact values of the distances (i.e. 26.84 m would be represented by the value 3).

From our data, we have 18 route segments in which a change of direction occurred. We thus have 18 data points from which to, using linear regression, fit a linear function able to predict the average confidence scores that were given for a segment. To avoid over-fitting we are using 3-fold cross-validation (i.e. a model is trained using 2/3 of the data, and evaluated on the remaining 1/3; this process is repeated 3 times). The three models all show a high correlation between the seven independent variables (the line of sight) and the dependent variable (the rating); for the overall model (using all the data) we obtain a correlation coefficient of 0.91. The mean absolute error is 0.73. The overall model we obtain for all data points is the following equation:

$$y = 4.82 - 0.16x_1 - 0.16x_2 + 0.07x_3 + 0.24x_4 - 0.16x_5 - 0.04x_6 + 0.01x_7$$

In the example in Fig. 5, a route segment from Route I, arrow A indicates the follower's current bearing. This arrow corresponds to 0° in our representation of the geographical context. The next route segment is reached via a right turn, but in this roundabout situation, other right turns are possible. In our experiment, the participants have given an average rating of 3.0 for the instruction "turn slightly right". The input vector for this particular situation is (8, 5, 8, 5, 7, 7, 8), and our regression model predicts a score of 3.18. Note that the average confidence scores do not span the whole range between 1 and 5. The lowest average for a segment is 2.8, the highest is 5.0. A score of 3 is thus a low score, meaning that in this route segment, a relative direction is not to be preferred.

Consider by contrast the situation depicted in Fig. 6. Here, participants gave an average rating of 5.0 for the relative instruction, which is not surprising given the simple street configuration. The input vector for this particular situation is (5, 4, 8,

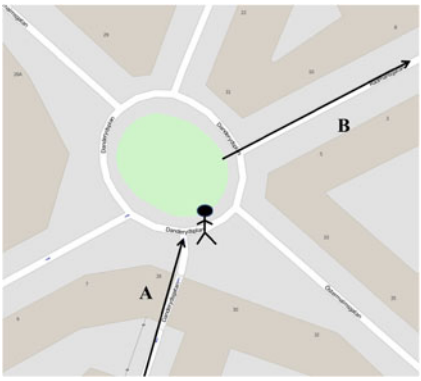


Fig. 5 An example route segment from Route I

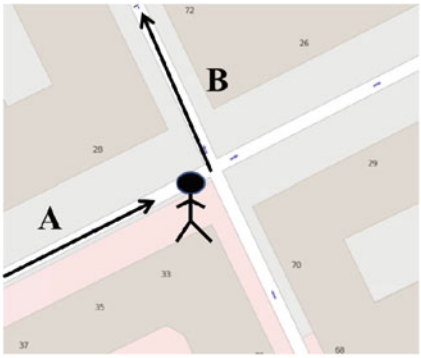


Fig. 6 An example route segment from Route II

8, 5, 3, 8), and the model predicts a value of 5.02. Here, an automatic system could confidently generate an instruction like "Turn left here".

7 Conclusion

Our data collection shows that different kinds of instructions do not work equally well at all decision points. In particular, instructions that avoid a landmark and use only a relative direction like "left" or "right", seem to be preferred at some decision points, particularly those with a simple configuration where streets meet at right angles.

We are suggesting a linear regression model to predict automatically whether such an instruction will be easy for a follower to interpret. The information we are using for this prediction is easily obtainable, e.g. from a database like OpenStreetMap, making it feasible to compute the confidence measure in real time. It is much easier to calculate the relative direction from a map than deciding which landmark to refer to and we are planning to include this automatic computation into

our direction-giving system. If the predicted confidence score is at least 4, the system can be reasonably sure that the relative direction will be interpreted in the desired way.

References

Albore, A., Boye, J., Fredriksson, M., Götze, J., Gustafson, J., & Königsmann, J. (2013). *Final pedestrian behaviour component*. Project deliverable, Spacebook EU 7th framework project 270019.

Allen, G. L. (1997). From knowledge to words to wayfinding: Issues in the production and comprehension of route directions. *Spatial Information Theory. A Theoretical Basis for GIS* (363–372).

Allen, G. L. (2000). Principles and practices for communicating route knowledge. *Applied Cognitive Psychology, 14*, 333–359.

Boye, J., Fredriksson, M., Götze, J., Gustafson, J., & Königsmann, J. (2014). Walk this way: spatial grounding for city exploration. In *Natural interaction with robots, knowbots and smartphones* (pp. 59–67). New York: Springer.

Caduff, D., & Timpf, S. (2005). The landmark spider: Representing landmark knowledge for wayfinding tasks. In *AAAI'05 Spring Symposium* (pp. 30–35).

Couclelis, H., Portugali, J. (Eds.). (1996). Verbal directions for way-finding: space, cognition, and language. *The Construction of Cognitive Maps*, *32*, 133–153.

Dahlbäck, N., & Jönsson, A. (1989). Empirical studies of discourse representations for natural language interfaces. In *Proceedings of European Chapter of the Association for Computational Linguistics* (pp. 291–298).

Daniel, M.-P., & Denis, M. (1998). Spatial descriptions as navigational aids: A cognitive analysis of route directions. *Kognitionswissenschaft, 7*, 45–52.

Denis, M. (1997). The description of routes: A cognitive approach to the production of spatial discourse. *Current Psychology of Cognition, 16*, 409–458.

Götze, J., & Boye, J. (2013). Deriving salience models from human route directions. *Workshop on Computational Models of Spatial Language Interpretation and Generation* (*CoSLI-3,* pp. 36–41).

Haklay, M., & Weber, P. (2008). OpenStreetMap: User-generated street maps. *Pervasive Computing, IEEE, 7*, 12–18.

Hund, A. M., & Minarik, J. L. (2006). Getting from here to there: spatial anxiety, wayfinding strategies, direction type, and wayfinding efficiency. *Spatial Cognition and Computation, 6*, 179–201.

Janarthanam, S., Lemon, O., Liu, X., Bartie, P., Mackaness, W., Dalmas, T., & Götze, J. (2012). Integrating location, visibility, and question-answering in a spoken dialogue system for pedestrian city exploration. In *Proceedings of SIGDIAL*, South Korea.

Li, R., Fuest, S., & Schwering, A. (2014). The effects of different verbal route instructions on spatial orientation. In *17th AGILE Conference on Geographic Information Science*.

May, A. J., & Ross, T. (2006). Presence and quality of navigational landmarks: Effect on driver performance and implications for design. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 48*, 346–361.

Michon, P.-E., & Denis, M. (2001). When and why are visual landmarks used in giving directions? *Spatial Information Theory, 2205*, 292–305.

Raubal, M., & Winter, S. (2002). Enriching wayfinding instructions with local landmarks. *Geographic Information Science, 2478*, 243–259.

Rehrl, K., Leitinger, S., Gartner, G., & Ortag, F. (2009). An analysis of direction and motion concepts in verbal descriptions of route choices. *Spatial Information Theory, 5756*, 471–488.

Richter, K.-F. (2008). *Context-specific route directions—generation of cognitively motivated wayfinding instructions*. DisKi 314/SFB/TR 8 Monographs (Vol. 3).

Striegnitz, K., Denis, A., Gargett, A., Garoufi, K., Koller, A., & Theune, M. (2011). Report on the second challenge on generating instructions in virtual environments (GIVE-2.5). In: *13th European Workshop on Natural Language Generation* (pp. 270–279).

Tom, A., & Denis, M. (2004). Language and spatial cognition: Comparing the roles of landmarks and street names in route instructions. *Applied Cognitive Psychology, 18*, 1213–1230.

Tom, A. C., & Tversky, B. (2012). Remembering routes: streets and landmarks. *Applied Cognitive Psychology, 26*, 182–193.

Labeling Streets Along a Route in Interactive 3D Maps Using Billboards

Nadine Schwartges, Benjamin Morgan, Jan-Henrik Haunert and Alexander Wolff

Abstract We consider the problem of labeling linear objects, such as streets, in *interactive 3D maps*, where the user can continuously pan, zoom, and rotate a perspective view of the scene. We dynamically annotate streets that belong to a user's route, assuming that the future course of the route, within the currently visible part of the map, is known or well predicted. We use *billboards* as annotations, that is, each label is a rectangle holding the annotation text, is oriented towards the user, placed at some distance above the midpoint of the street to be labeled, and connected to the point by a vertical line segment, the *leader*. Our goal is to maintain an overlap-free labeling that reacts to changes of the view in real time. To this end, we dynamically vary the lengths of the leaders. In order to achieve that labels move smoothly, we do not strictly forbid label–label overlaps. We present a force-directed algorithm that applies forces to labels to cause overlapping labels to repel each other, while keeping leaders as close to their desired length as possible. On real-world data, with a realistic number of labels, we obtain frame rates of more than 400 frames per second, while drastically reducing the total overlapped area per frame, compared to an algorithm with fixed leader lengths.

Keywords Dynamic maps · Interactive maps · Map labeling · Street labeling · Billboards

N. Schwartges (✉) · B. Morgan · A. Wolff
Lehrstuhl für Informatik I, Universität Würzburg, Würzburg, Germany
e-mail: nadine.schwartges@uni-wuerzburg.de

B. Morgan
e-mail: benjamin.morgan@stud-mail.uni-wuerzburg.de

J.-H. Haunert
Institut für Geoinformatik und Fernerkundung, Universität Osnabrück, Osnabrück, Germany

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_16

1 Introduction

Interactive maps commonly allow users to zoom, pan, and rotate the map. Examples of such maps are Google Maps[1] or digital devices with navigation software. Some interactive maps provide only a two-dimensional (2D) view whereas others, for instance, Google Earth,[2] provide a perspective 3D view. Such tools help users to orient themselves in an unknown environment.

Mobile devices with interactive maps commonly provide a *map mode* with which users can freely move about (literally, users travel with their fingers on the map). Additionally, the same devices offer a *navigation mode*, a route planner, that leads users from their current locations to specified destinations. Sometimes, it is even possible to interact with the map in navigation mode. To aide users in orienting themselves, independent of the mode, map objects are usually annotated by textual or pictorial *labels*.

There are three types of objects represented in maps: *points*, such as a cities (in small-scale maps) and points of interest; *lines*, such as streets and rivers; and *areas*, such as countries and lakes. In this work we consider the problem of labeling streets, but our results can also be used for labeling point features (without any changes) or area features (assuming that we are given a suitable point in each area).

In printed large-scale maps, streets are commonly labeled *embedded*, that is, the label is placed inside the area occupied by the street and follows the curvature of the street. In interactive 3D maps, embedded labels are rendered with perspective distortion, as in Fig. 1a, or they are rendered parallel to the view plane, as in Fig. 1b. Sometimes, labels are placed straight—neglecting the exact course of the street—as in Fig. 1c, in order to save computation time and to improve the readability of the label text. It is also quite common to make use of *billboards*; examples can be found in some built-in car navigation systems. Our billboards consist of (i) a *label*, that is, a rectangle that is oriented towards the user and holds the *label text*, and (ii) a *leader* that connects the point to be labeled, the *reference point*, with the label. The leader of a billboard can be a line, as in Fig. 1d, or a more complex object, such as a triangle or arrow. This paper focuses on placing billboards on streets in interactive maps that are in navigation mode.

When the navigational device leads a user to a destination, the route which the user has to follow is usually highlighted. We call this route and the corresponding streets *active*. Accordingly, we refer to streets that are not contained in the active route as *inactive*. By using billboards instead of embedded labels for the active route, we highlight active streets. Moreover, horizontally oriented text can be read faster than rotated text (Larson et al. 2000; Wigdor and Balakrishnan 2005). We thus improve the readability of those labels that are, at any given time, the most important ones for the user. For labeling the remaining streets, our algorithm can be

[1] https://maps.google.com/, accessed Oct. 1, 2014.

[2] https://earth.google.com/, accessed Oct. 1, 2014.

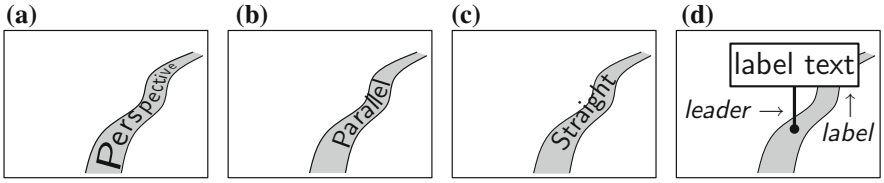


Fig. 1 Different ways of labeling streets in a large-scale 3D map. Our approach uses billboards. **a** Perspective, **b** parallel, **c** straight and parallel and **d** billboard

combined with our algorithm that embeds labels into streets in large-scale interactive maps (Schwartges et al. 2014).

Our Model We consider a dynamic scenario where the user follows a route in an interactive navigation mode. To this end, we consider a time interval [0,T] in which either the navigational device automatically manipulates the map, or the user manually interacts with it. During this time interval, the content of the display of the navigational device is redrawn repeatedly; the content between two updates is a *frame*. Accordingly, we discretize the time interval into a sequence $t_1, t_2, \ldots, t_k$ (with $t_1 = 0 < t_2 < \ldots < t_k = T$) of points in time that correspond to frames. At any given time t_i, the user can see a trapezoidal region R_i of the map. This is the image of a projection (whose center is the camera) of the rectangular, vertical screen on the horizontal map. When panning, R_i is translated on the map; when zooming, R_i is scaled; when rotating, R_i is rotated; and when changing the camera angle of the 3D view, we transform R_i perspectively, more precisely, if we change the camera angle such that we can see more of the map at the horizon, the edge of R_i that corresponds to the bottom edge of the view gets shorter, the other base edge gets longer, the two angles at the shorter base edge of R_i gets larger, and the legs get longer; and vice versa.

For our model, we lean on the physical principle of a thermodynamic equilibrium. We assume that in each frame there might be a label–label overlap, because either a new label has come into the view, or an overlap of the preceding frame has not been completely solved, or a solved overlap from the preceding frame has caused another overlap. Each label–label overlap induces a force F_{overlap}. Moreover, we define a desired leader length. A leader that is too long or too short induces a force F_{leader}. As we assume that we only know the currently visible part of the map, we solve overlaps only from one frame to the next. We can translate the goal of establishing an overlap-free labeling in one frame to the goal of minimizing the acting force

$$F = \sum \left|F_{\text{overlap}}\right| + \sum \left|F_{\text{leader}}\right|$$

in that frame. More precisely, we aim to minimize each individual force, as some forces of the same label might cancel each other out, even though there is still a label–label overlap (which happens, for example, if a label A pushes label B down,

but B is also pushed up by the force from having too short a leader). With the application of a physical model, we expect the movements of labels to look natural —as if they were subjected to physical laws.

Our Contribution We start with a motivation why it is reasonable to design an algorithm for placing billboards to streets (see Sect. 3). To this end, we present the results of a survey asking for the aesthetic and practicability of such labelings. Next, we present our force-directed algorithm for labeling streets with billboards in an interactive navigation mode (see Sect. 4). Throughout the navigation, the algorithm maintains an aesthetic and practical labeling; it uses repelling forces for resolving overlaps and both attracting and repelling forces for keeping leaders as close to their desired length as possible. All label movements are smooth. Tests of the implementation of the algorithm on real-world data show that our algorithm yields interactive frame rates of more than 400 FPS (see Sect. 5). A video that shows our algorithm in action is available at http://lamut.informatik.uni-wuerzburg.de/dynaroutelab.html.

2 Related Work

In his seminal work on label placement (for printed maps), the Swiss cartographer Imhof (1975) established many rules for good label placement. His two most important rules are that labels should be legible (R1) and always yield a correct label–object association (R2). We fulfill these rules since we align labels horizontally, we connect the label and the reference point by a leader, and we avoid overlaps.

Imhof also says that a label should reflect its object's importance (R3). In our setting, where we want to label streets on the active route, Imhof's rule is automatically fulfilled since our perspective view draws the closer (and, hence, in that moment more important) labels in the foreground larger than the distant (and less important) labels in the background. Further, Imhof states that labels should occlude the map background as little as possible (R4). Currently, we do not take this rule into account. This can, however, be achieved by making labels semi-transparent (at the cost of reducing legibility). In order to avoid label clusters, Imhof suggests carefully selecting the objects to be labeled (R5). In our navigation-mode scenario, we simply select the next n streets on the route to be labeled. Among these, we show all labels that fall into the current view; we avoid clusters by changing the leader lengths.

For drawing graphs aesthetically, Eades (1984) introduces an algorithm which is based on a physical model using forces. He considers a drawing aesthetic if the edges of the graph have similar length and the graph is as symmetric as possible. To this end, the vertices of the graph may move in any direction. Adjacent vertices are supposed to keep a certain distance from each other, non-adjacent vertices repel each other. In our model, by contrast, the reference point of a label is fixed and the

Table 1 Our approach compared to some related work

	Interaction types	History	Computation time	Mode	Technique
Vaaraniemi et al.	Pan, zoom, rotate, 3D	Considered	5.5 ms per update	Map	Force-based
Maass and Döllner	*Unknown*	With workaround	"Real time"	Map	ILP, greedy
Gemsa et al.	Pan, rotate	Considered	sec to min	GPS	Greedy
Our approach	Pan, zoom, rotate, 3D	Considered	>400 FPS	GPS	Force-based

GPS navigation mode, *FPS* frames per second, *sec* seconds, *min* minutes, *greedy* greedy algorithm, *ILP* integer linear program

point where leader and label touch can move only vertically. Similar to the edges in Eades' approach, our leaders try to have a certain length. The author states that he does not use Hooke's law (as we do) but a logarithmic function to obtain the edge lengths because a logarithmic function works better for vertices that are far apart. Eades' algorithm computes the forces and thus the new positions of the vertices several times. Our approach is also iterative: in each frame, we recompute the lengths of the leaders if the corresponding labels overlap or are not at their desired length.

Table 1 gives an overview about our algorithm and some of the related work that we discuss in the following.

Vaaraniemi et al. (2012) give a force-directed algorithm that is, to some extent, similar to ours. Their algorithm labels points and areas horizontally. Depending on the current perspective, a street label is either placed horizontally or is aligned to a straight line that approximates the course of the street. In contrast to our approach, their algorithm operates in what we call map mode and they allow any leader direction. The main difference between the two approaches is that while we always display all labels that fall into the view, Vaaraniemi et al. remove labels in two situations, namely if a label moves too fast or if a label is overlapped such that the forces acting on it cancel each other. While we label only the active streets, Vaaraniemi et al. label all types of objects within the view. As they resolve overlaps by moving and selecting labels, we expect a lot of changes on the screen, which may be distracting (for example, for a car driver using a GPS). In terms of speed, Vaaraniemi et al. report that their algorithm computes the layout for 512 objects within 5.5 ms (this corresponds to 180 FPS *without* rendering).

Gemsa et al. (2013) investigate the off-line version of a point-labeling selection problem (in 2D) with respect to an active route. They assume that the entire active route is given in advance and fixed (while it may change in our case). They do not use leaders, but they assume that each label has a fixed position relative to the point it labels. As in our setting, they have a fixed camera above the active route and a map that turns and translates such that the direction of movement appears to be upwards/North. Each label may be visible during several intervals. In order to

reduce flickering, for each of these intervals, Gemsa et al. allow for selecting a *connected* (sub)interval in which the corresponding label is finally visible. The authors aim for an overlap-free labeling for the entire route that maximizes the total length over all selected (sub)intervals.

The authors show that their problem is NP-hard. They present an exact algorithm (an integer linear program that solves the problem optimally; in exponential time). They test their algorithm on 1000 active routes at three different scales. On average, they need less than a second for optimizing the labeling of routes with about 162 labels and less than six seconds for routes with about 313 labels. A few routes, however, took them several minutes. The authors also give efficient approximation algorithms, but do not include any test results.

Maass and Döllner (2006) place billboards in interactive 3D maps that also allow for 3D objects (such as buildings). They require that the further away the labeled object is from the user, the smaller the label and the higher the leader. Their algorithm subdivides the view plane into a grid and places labels incrementally. Each placed label blocks several surrounding grid cells for other labels. The algorithm does not consider the *history* of the labeling, that is, the labeling of frame f_i does no take the labeling of frame f_{i+1} into account but computes a new labeling from scratch. In order to avoid jumps that might confuse the user, the labeling does not change while the user interacts with the map. If the user stops interacting, labels smoothly move to their new positions. In contrast, we immediately react to user interaction while taking into account the labeling of the preceding frame.

We also pursue another concept to label streets: we introduce an incremental algorithm for placing embedded labels into streets in interactive maps (Schwartges et al. 2014). Our aim is to maintain an overlap-free labeling where as many streets as possible are labeled. We avoid placing labels on street parts that contain strong bends. To this end, we introduce a cost function with which we evaluate any interesting label position on each of the visible streets. Such an embedded labeling complements the labeling of active streets.

Maass et al. (2007) present the results of a user study dealing with labelings in 3D maps using billboards. The authors examine the problem of leaders inducing wrong depth cues; for instance, a leader, whose reference point obviously lies behind a building, is drawn over the building. Most of the participants of the user study judge depth cues that are accurate as more comfortable. The authors finally suggest introducing a parameter that measures the perceivable perspective disturbance. If the parameter is applied in labeling algorithms, it is intend to improve the label placement, but, simultaneously, it sometimes permits wrong depth cues. In our approach, we always aim for correct depth cues.

Similarly, Vaaraniemi et al. (2012) describe an expert study. The first result is that labels in a 3D view should shrink with distance to the user, in order to create a better understanding of the depth. We take this result of their expert study into account in our implementation.

Moreover, Vaaraniemi et al. ask the experts if streets should be labeled embedded, straight-line aligned, or horizontally. Four of six experts judge horizontally-placed labels as very legible, although they also note that a high search time is required to

associate a label with its object. Again, we follow this judgement by using horizontal labels; we improve the label–object association by using leaders. On the other hand, five of six experts like the embedded labeling, but also point out that if labels are situated in strong bends, they might be badly readable. Three of six experts observe that embedded labels yield a good label–object association.

3 Survey

We conducted a small survey in order to decide which variant of our concepts to implement first. In total, we had 19 participants (one female), aged between 19 and 26 years, all of them studying a technical subject. We asked the participants which of the labelings shown in Fig. 2 they like most and which of them they think is the most practicable for navigation systems. Moreover, we asked which of the labelings shown in Fig. 3 they like most. We found out that 47 % preferred the labeling in

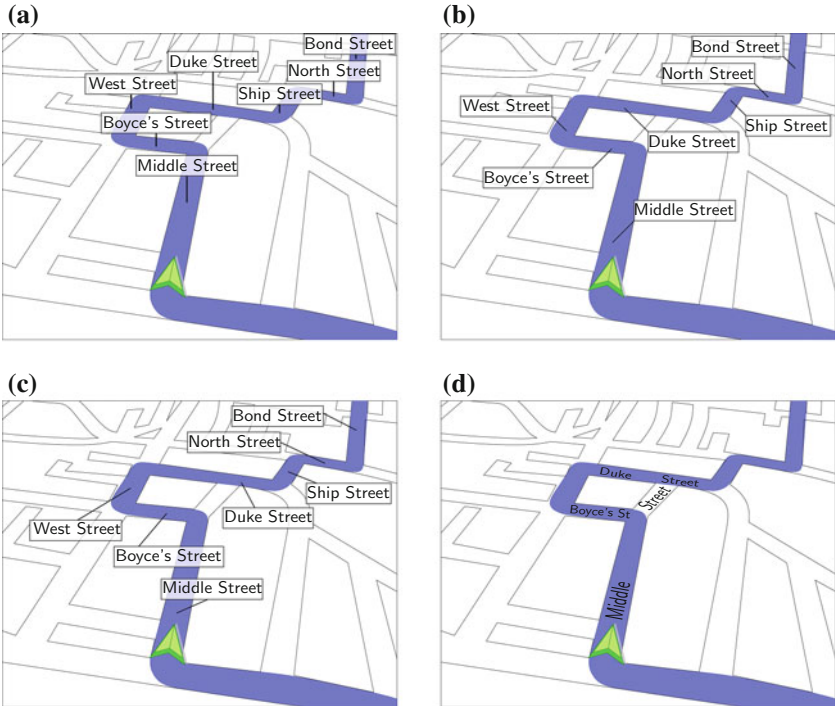


Fig. 2 Figures similar to those we have shown the participants of our survey. (We do not have permission to publish the original figures. They differ in that we removed the map background and the embedded labels of inactive streets). **a** Vertical leaders, **b** leaders are rotated by $k \cdot 45°, k \in \mathbb{N}$, **c** arbitrarily-rotated leaders and **d** embedded labels

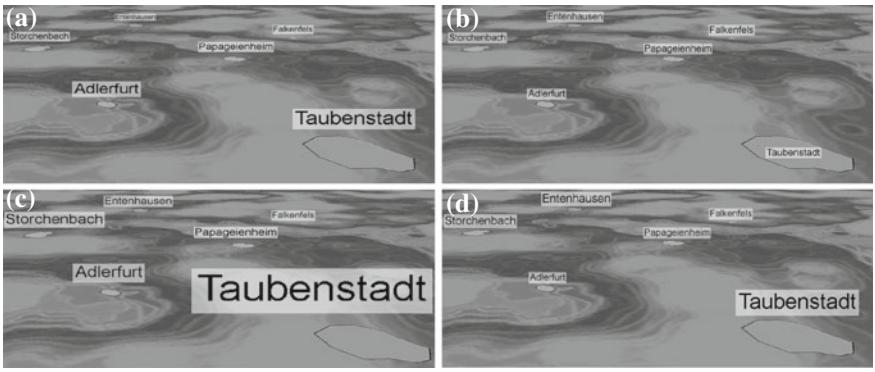


Fig. 3 Figures we showed to the participants of our survey. **a** Constant size in world space, distance dependent in screen space, **b** constant size in screen space, **c** population dependent in world space, distance dependent in screen space and **d** population dependent in screen space

Table 2 Results of our survey

	Aesthetics (%)	Practicability (%)
Figure 2a	32	32
Figure 2b	21	11
Figure 2c	0	5
Figure 2d	47	53

	Aesthetics (%)
Figure 3a	47
Figure 3b	16
Figure 3c	0
Figure 3d	37

Fig. 2d and that this solution is also considered most practicable by 53 %. The next best rated possibility was Fig. 2a with 32 % for both questions. As we have already published our work about labeling streets in interactive maps using embedded labels (Schwartges et al. 2014), in this work, we tackle the problem of labeling streets using billboards. To conclude, 47 % preferred the labeling in Fig. 3a. We give the remaining numbers in Table 2.

4 Algorithm

In this section, we propose a simple force-directed algorithm for preventing labels from *occluding*, that is, overlapping, other labels. There are several auxiliary algorithms that are needed: setting up the environment, reading and routing through a street map, and so on; we will not cover these algorithms in depth.

Fig. 4 A street network with a route and a pointer

Consider a street map which contains an active route from a starting location A to a destination location B. The active route is a sequence of *street polylines*, where each street polyline is only a subsection of the entire street, in the sense that one traverses a street for only so long as to reach the intersection that leads to the next street. We shall use the terms street polyline and street interchangeably. The route is traversed by a *pointer* π which represents the user, and is typically modeled as a triangle or vehicle (see Fig. 4). The camera is placed at some distance behind the pointer.

Our goal is to label the individual streets and prevent labels from occluding each other. We place labels (or rather billboards) in world space, each above the reference point of its street. For the sake of simplicity, we assume that each street has only one reference point, namely at the midpoint of the polyline. Hence we label a street only once, regardless of its length. (It is of course not difficult to overcome this restriction.) Recall that we connect each label to its reference point with a vertical leader whose length can dynamically vary. We denote by h_0 the *default leader height*, which is the desired height. A label can move only vertically, by extending or contracting the leader. To simplify further discussion, let us consider each billboard as a complete entity, denoted by λ_i, where i denotes the i-th reference point on the route. The actual height of the leader of the billboard λ_i at any given time t is denoted by $h_i(t)$.

Given the route R, let $N := |R|$ denote the number of streets along the route. It makes little sense to display every label for every street in the route at the same time, as the user is primarily interested in the next few streets ahead. Therefore, we upperbound the number of *placed* labels at any time to a constant $n \leq N$. Let $I = \langle l, \ldots, m \rangle$ denote the queue of currently placed labels; we have $|I| \leq n$. (We have $|I| < n$ if the remaining part of R consists of fewer than n streets.) When the distance of the pointer π from the reference point q_l of label λ_l falls below a threshold ε, then label λ_l is dequeued from I, and label λ_{m+1} (if it exists) is enqueued. Note that the number n of placed labels is sometimes greater than the number of *visible* labels, that is, the labels within the view. We also place labels that lie outside the view, in addition to visible labels, for two reasons. First, we can elide checking if a label is about to enter the view. Second, when labels do enter the view, they do not disturb the visible labeling much, as they have already been considered by the algorithm.

4.1 Force-Directed Approach

We propose a simple force-directed algorithm. Forces are exerted on each label by other labels and by its own leader; these forces cause the label to move in a way minimizing the aggregate force. The leader acts as a spring, keeping the label close to its reference point, while the labels repel each other much in the same way that same-pole magnets do. Excess aggregate force is mapped to a change in leader height. To prevent a label from oscillating strongly between two other labels, the force is scaled by a temperature, which is reduced when the aggregate force changes its *sign* from one iteration to the next.

While the labels themselves live in 3D *world space*, we see them in a projection onto 2D *screen space* (see Fig. 5). Certain calculations, such as determining whether two labels overlap, are therefore performed in the screen space. The screen-space representation of a label is a rectangle; we exclude the leader from this rectangle. We determine if two labels overlap by inspecting whether their screen-space projections overlap. To ease further discussion, we refer to the label in screen space simply as the label projection and again denote it by λ_i.

One last note before we move on: we say the algorithms in this section are *frame-based*. In each new frame, pointer and camera may have moved, labels may be seen from a different perspective, labels may have been moved up or down by their leaders, auxiliary algorithms take the new data into account, and the force-directed algorithm runs. Frames may coincide with rendered frames or they may be timer-based. The only requirement is that changes in leader height are reflected in the screen-space projection in the next frame. Since the information the algorithms need is primarily for the current frame, we elide the time t, so that for frame-bound value, say $h_i(t)$ for example, we write $h_i := h_i(t)$.

In the following, we discuss how the various forces are computed and how they change the leader height.

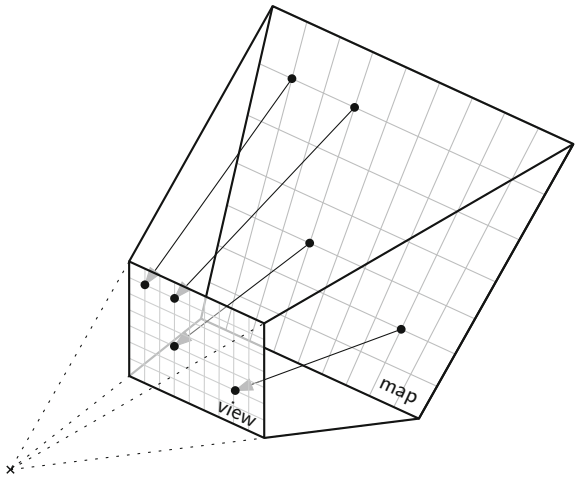


Fig. 5 Projecting points lying in the currently visible part of the map from world space to screen space

4.2 Spring Force

The leader of a label is modeled as a simple tension spring, which has a default height of h_0. A spring can undergo extension as well as contraction, which is negative extension. The force is given by Hooke's law, and is simply a spring constant k multiplied with the leader extension:

$$F_i^s := -k \cdot (h_i - h_0). \tag{1}$$

Due to the way the spring force flows into the overall force acting upon a label, the main effect k has is to affect the speed at which labels return to their default height: the greater k is, the stronger the force.

4.3 Aggregate Repulsive Force

Principally, the aggregate repulsive force F_i^r for the label λ_i consists of the sum of all repulsive forces $r(i,j)$ between λ_i and every other placed label λ_j:

$$F_i^r := \zeta \cdot \sum_{j \in I \setminus \{i\}} r(i,j). \tag{2}$$

This is a slight simplification, as we shall see in Sect. 4.4, but for initial understanding it is sufficient. The constant ζ serves to weight the aggregate repulsive force.

The function r relies on several concepts we will introduce first: the relevance of labels, the sign of labels, the distance metric, and the interplay between labels.

Sign Let $\mu(i)$ be the midpoint of the projection of λ_i and let $\mu_y(i)$ be the y-coordinate of $\mu(i)$. Let the sign $\sigma(i,j)$ of a label λ_i denote whether it is above or below another label λ_j, $i \neq j$:

$$\sigma(i,j) := \begin{cases} -1 & \text{if } \mu_y(i) < \mu_y(j), \text{ and} \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

We can categorize each λ_j in one of three categories in regard to λ_i: either the midpoint of λ_j is below, above, or at the same height as that of λ_i. The function σ however, handles only two cases. One might ask: might not two labels, in case of midpoint equality, move upwards at the same rate and thus arbitrarily far from their reference point? We think this cannot happen, because multiple factors affect the change in height, and these factors cannot all be the same, unless $\lambda_i = \lambda_j$, which we disallow. Hence one label shall travel further, and in the next frame the midpoints are different.

Distance Metric We can model the repulsive force between two labels as a function of their distance to one another. This is an attractive model, as it is easy to reason about, but it leaves us with the problem of defining a useful distance metric δ.

Not all labels need repel each other. We have already ascertained that labels have only one degree of freedom, namely in the vertical axis (y-axis). It is reasonable therefore to allow labels to only affect each other when a movement would make sense. This occurs only when labels intersect in the horizontal axis (x-axis). In our distance metric then, labels that do not intersect in the horizontal axis have a distance of ∞.

What when the labels overlap? Distance between typical objects is measured starting from 0, when the objects are right next to each other. To deal with this, we can redefine a distance unit to equal the height of a label projection. Let $\omega(i,j) \in [0,1]$ denote the percentage that λ_i is occluded by λ_j, that is, $\omega = 1$ iff λ_i is completely covered by λ_j. Note that ω is asymmetric. Further, let $\gamma(i,j)$ denote the y-offset between λ_i and λ_j, in the case that they do not overlap. This leads us to the following definition of δ:

$$\delta(i,j) := \begin{cases} \infty & \text{if } \lambda_i \text{ and } \lambda_j \text{ do not intersect in } x\text{-axis,} \\ 1 - \omega(i,j) & \text{if } \omega(i,j) > 0, \text{ and} \\ 1 + \gamma(i,j)/h_i & \text{otherwise.} \end{cases} \tag{4}$$

Note that this definition is somewhat inconsistent, because $\delta \in [0,1]$ is a measure of area, while $\delta > 1$ a measure of height. This is intentional. If two labels overlap by one pixel in the horizontal axis, they cannot by the above Eq. 4 have a distance of 0. This behavior is useful, as it differentiates between full and partial occlusions.

Resulting Force When the label λ_i is fully occluded by λ_j, we let the force be a constant F_{limit}; when $\delta = 1$, we let the force equal F_i^s. The force grows quadratically for $\delta < 1$, and linearly when $\delta \in [1,2]$. Finally, we restrict that labels that have $\delta > 2$ do not affect each other. Any label farther away need not have an effect, and if it should come closer, then at some point $\delta \leq 2$ will hold. We describe the entire algorithm that computes the resulting force $r(i,j)$ that λ_j exerts upon λ_i in Algorithm 1.

Algorithm 1: Resulting Force

input : labels λ_i and λ_j
output: force that λ_j exerts upon λ_i

$d \leftarrow \delta(i,j)$
if $d = 0$ **then**
  return $\sigma(i,j) \cdot F_{\text{limit}}$
else if $d \leq 1$ **then**
  $v \leftarrow r \cdot \left(1/\delta(i,j)^2 - (1 - F_i^s)\right)$
  if $v > F_{\text{limit}}$ **then**
    return $\sigma(i,j) \cdot F_{\text{limit}}$
  else
    return $\sigma(i,j) \cdot v$
else if $d \leq 2$ **then**
  return $\sigma(i,j) \cdot F_i^s \cdot (1 - \delta(i,j))^2$
return 0

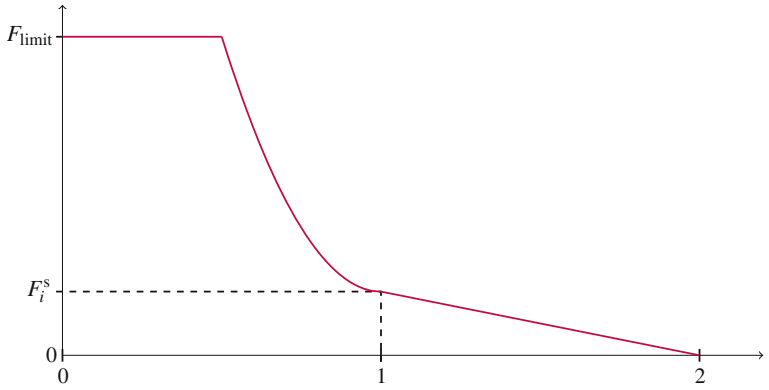


Fig. 6 The graph of function r, with the distance between two labels in the x-axis

The function r is monotonic and continuous, despite potentially flattening out before $\delta = 0$. Figure 6 shows the graph of r. We could define the function to approach F_{limit} when $\delta \to 0$, however it would provide only questionable benefit while requiring more computational power.

This function definition has the fortunate property that the force acting upwards on label λ_i from a lower label λ_j will equal the spring force pulling λ_i down precisely when λ_i is right next to λ_j. This provides just the right amount of force to prevent occlusions, while allowing labels to return to their default height when possible.

4.4 Aggregate Force

Intuitively, the overall force F_i acting on a single label λ_i consists of the sum of the repulsive forces F_i^{r} (with constant factor ζ) and the spring force F_i^{s}:

$$F_i := F_i^{\text{s}} + F_i^{\text{r}}. \tag{5}$$

This turned out to be too simplistic. What happens to λ_2 when three labels λ_1, λ_2, and λ_3 are stacked? If λ_1 exerts the same force on λ_2 as λ_3 does, then the forces cancel each other out and the remaining force is F_2^{s}, which causes λ_2 to descend into λ_3, thereby causing a collision.

Hence we need to keep positive forces F_i^+ and negative forces F_i^- separate, so that we can handle this case specially:

$$F_i^+ := \sum\nolimits_{j \in I \setminus \{i\},\, r(i,j) > 0} r(i,j)$$
$$F_i^- := \sum\nolimits_{j \in I \setminus \{i\},\, r(i,j) < 0} r(i,j).$$

Since if $r(i,j) = 0$ it affects nothing, we can ignore it in the above definitions. With now two sets of repulsive forces, we include F_i^s iff at least $F_i^+ = 0$ or $F_i^- = 0$. The formula is now

$$F_i := \begin{cases} F_i^s + F_i^+ + F_i^- & \text{if } F_i^+ = 0 \text{ or } F_i^- = 0, \text{ and} \\ F_i^+ + F_i^- & \text{otherwise.} \end{cases} \tag{6}$$

If a label is surrounded by two labels, the spring force is thus ignored.

4.5 Temperature

For each label λ_i, we define a temperature T_i that acts as a factor in scaling the force. The default temperature is defined by a constant T; this is the temperature that labels have when first placed. We track the aggregate force of the label between two frames; if the sign of the force changes, we reset the temperature to a constant T_{base}. To prevent two labels from jumping too quickly away from each other, a negative force (pushing the label downwards) counts as a sign change from a force of 0. If the sign remains the same, we multiply the temperature by a constant T_{step}, thereby increasing the temperature slightly. In order to prevent the temperature from becoming arbitrarily large, we limit it to a constant T_{limit}. It is assumed that $T, T_{\text{base}} > 0$ holds.

4.6 Leader-Height Change

The combination of aggregate force and temperature let us derive either an extension or contraction Δ_i of the leader of label λ_i, which we can (again) scale by a constant factor χ:

$$\Delta_i := \chi \cdot T_i \cdot F_i. \tag{7}$$

Because of rounding error, it is unlikely that F_i will ever reach an equilibrium between different forces. To prevent labels from always moving, we only apply a leader-height change when F_i is greater than a constant F_{min}.

4.7 Complexity and Runtime

The setup time complexity for all the algorithms is linear in the size of the input. We consider the complexity of the algorithm for a single frame. The auxiliary algorithms require at most $O(n)$ time, where n is the maximum number of placed labels in a frame.

The force directed algorithm considers each of the labels in I, and performs for each the following:

1. Calculates the spring force in $O(1)$ time (Eq. 1).
2. Calculates the repulsive force in $O(n)$ time (Eq. 2 and Algorithm 1).
3. Calculates the leader change and applies it in $O(1)$ time (Eq. 7)

Thus we have a runtime of $O(n^2)$ in total. Since we control the value of n, we can determine the maximum runtime of the algorithm in each frame. This is especially useful for embedded applications. In practice, letting $n = 10$ seems sufficient for us and has next to no negative impact on performance. As we show in our experiments, even with a quite large n, we obtain interactive frame rates.

4.8 Implemented Improvements

In our description of the force-directed algorithm, we constrained ourselves to the essential essence of the algorithm. Here we present two additional modifications. None of the modifications change the complexity of the algorithm.

Margins In our problem definition, labels must, at least, not overlap. Sometimes it is desirable that they maintain a margin of separation from each other. To this end, we give each label a bottom margin of v by extending the screen-space projection of the label by v units (for example pixels). This margin is not visible to the user.

Relevant Labels The queue I of placed labels is maintained by an auxiliary algorithm. However, there are labels for any given frame which are not necessarily relevant to the force-directed algorithm or even might cause undesired behavior. We remove such labels from I. Any label that is removed has its values reset to its defaults. Finally, in each iteration, we use $\hat{I} \subseteq I$ as the set of *relevant* labels; we define $n' := |\hat{I}|$.

Some labels may be so far away or out of sight, that to include them in force calculations would seem unnecessary (see Fig. 7a). We define the following optimization then, in which we effectively ignore from I any label that has an area less than ξ. The label is still rendered, but it is not considered in any force calculation. Further recall that our labels are placed in world space. To this end, some labels lie behind the camera (see Fig. 7b). This might cause problems when projecting them from world space to screen space. We remove labels lying behind the camera from

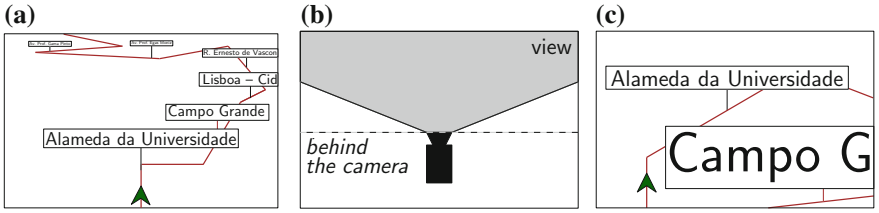


Fig. 7 Relevant and nonrelevant labels (*Note* these are no screenshots!). **a** Labels in the far back are too small to read, **b** labels behind the camera might cause wrong projections and **c** an adverse course of the route causes a very large label

I and we do not render them. Last, if a future part of the route is very near to the user, some labels become undesirably large (see Fig. 7c). Whenever the area of a label becomes larger than some value a, we remove the label from I and do not render it.

5 Experiments

We have implemented our force-directed algorithm from Sect. 4, a testing environment, and a *static* algorithm for comparisons. For our implementations and tests, we used C++ with OpenSceneGraph 3.1[3] and Boost C++ Libraries 1.56[4] on Linux 3.16 with a 2.5-GHz Intel dual-core processor, 8 GB of RAM, and an Intel HD3000 integrated graphics card. We applied the GCC-4.9.2 compiler to produce 64-bit binaries with compiler optimizations. For the testing environment, we used an OpenStreetMap data set provided by Geofabrik[5] from which we extracted the downtown street network of Würzburg, a town of 120,000 inhabitants in southern Germany. In order to simulate the navigation mode, we determined three different routes through Würzburg along which we created a camera path each. We tested $n = 10$, 25, and 50. We think, however, that $n = 10$ is the most reasonable value for the rather small displays of navigational devices. Our virtual navigation system had a resolution of 1366 × 768 pixels. In order to maintain n placed labels for each frame and to guarantee paths of equal lengths, we stopped the camera paths as soon as $|I| < 50$, that is, when the remaining part of the route had less than 50 labels left. In total, we placed $N = 69$, 96, and 131 labels in the first, second, and third path, respectively. (Note that number of actually placed labels is $N - 50 + n$.) The paths needed between 18s and 68s. In our tests, we only panned and rotated the map whereas we rotated 13 % of the total time. (We do not expect that the frame rate

[3] http://www.openscenegraph.org/, accessed Nov. 28, 2014.

[4] http://www.boost.org/, accessed Dec. 4, 2014.

[5] http://download.geofabrik.de/, accessed Nov. 28, 2014.

Fig. 8 Screenshots of our program in an artificial data set. *In reading direction*: Within 1.5 s, the overlap of the initial label placement is resolved. On the very first subfigure, the total overlap is 6773 pixels

drops for the remaining interactions as there is no special handling for the different interaction types—neither in our algorithm nor for rendering.) In the camera paths for our tests, the pointer had a constant position and direction on the screen (as in Fig. 7). For rotations, the pointer stopped its drive and rotated; then it continued the drive.

Our set of configurable values yielding nice-looking, smoothly moving labelings is as follows: $h_0 = 5.0$ units in world space,[6] $k = 0.25$, $\zeta = 1.0$, $F_{\text{limit}} = 5.0$, $\chi = 0.2$, $\left(T, T_{\text{base}}, T_{\text{step}}, T_{\text{limit}}\right) = (1.0, 0.1, 1.05, 5.0)$, $F_{\min} = 0.1$, $\xi = 100$ pixels, $v = 5$ pixels, $\alpha = 0.25$ of the total resolution, $\varepsilon = 10.0$ units in world space.

For the static algorithm, we just fixed the leader lengths to h_0. We ran the same camera paths as for the dynamic labeling algorithm.

Figure 8 shows some screenshots of our algorithm. When we start the algorithm, the leader lengths are equal. In this example data set, the overlap resolves within two seconds. In Table 3, we summarize our results for the force-directed algorithm and the static algorithm, both applied to the real-world data set.

In order to evaluate the processing time of our algorithm, for each path and each value of n, we measured the number of totally drawn frames as well as the total running time. By these two values, we computed the *frame rate* as the number of drawn frames per second (*FPS*). Table 3 shows the frame rates for each value of n, averaged over the three different paths.

Our algorithm yields very good frame rates of more than 400 FPS when it places n = 50 labels or less. If we only render the active route, the billboards, and the pointer (in other words, we do not render the street network), the frame rate increases by about 54 FPS. On the other hand, Table 3 shows that the frame rates of the static algorithm are only better by a few frames compared to the force-directed

[6]For comparisons: we set the font size of the label text in world space to 3 units.

Table 3 Results of our experiments for the static and the force-directed algorithm

	Static			Force-directed	
	Frame rate	Overlap		Frame rate	Overlap
n	FPS	px	n'	FPS	px
10	453	3190	8	452	49
25	444	4930	19	444	91
50	435	5980	34	431	116

For the frame rate, we divided the number of frames by the running time in seconds; for the remaining values, we averaged the numbers over the number of frames. For each measurement, we assured that at least n streets were still ahead
n placed labels, n' relevant labels, *FPS* frames per second, *px* pixels

algorithm. This is due to the fact, that, for the static algorithm, we have stretched the limits of what OpenSceneGraph combined with the integrated graphics card can achieve. To verify this, we also tested for *one* map the frame rate while OpenSceneGraph idled: without any computations or rendering but with loading the map and the route, we reached frame rates of about 450 FPS at our system.

For each path and each value of n, we recorded also the number of overlapped pixels. We counted only the overlaps of relevant labels in the view. We divided the total number of overlapped pixels by the total number of frames. Table 3 shows that, compared to the static algorithm, we could reduce the number of overlapped pixels per frame by about 98 % by applying our force-directed algorithm.

Unfortunately, we cannot compare our results to Maass and Döllner (2006) as they only state that their algorithm "operates in real time". Similarly, we cannot compare to Gemsa et al. (2013) as they compute the entire labeling in advance, that is, the frame rate is determined by the rendering algorithm only but not by the labeling algorithm. Vaaraniemi et al. (2012) state that their algorithm has a frame rate of 180 FPS for computing label positions if they disable the rendering. If we only render the important part of our visualization, that is, the route, the labels, and the pointer, for $n = 50$, we obtain frame rates of almost 500 FPS. In general, however, a frame rate of 24 FPS is qualified fluid. Thus we conclude that our algorithm for placing billboards to active streets in an interactive navigation mode for 3D maps yielding frame rates of more than 400 FPS is highly real-time capable and it is really worth trying to combine it with an algorithm that labels the remaining streets embedded.

6 Conclusion and Future Work

We have introduced a force-directed algorithm for placing billboards with leaders of dynamically varying lengths to active streets in interactive 3D maps. In our approach, each reference point tries to keep its corresponding leader at a desired length; overlapping labels repel each other. From frame to frame, we minimize the

unbalanced forces. This yields labelings that avoid label–label overlaps of billboards that are near to the user but accepts overlaps in the background. Our algorithm directly reacts to changes of the current view by smoothly moving overlapping labels. In our tests on real-world data with a realistic number of labels, our implementation reached interactive frame rates of more than 400 FPS and reduced the number of overlapped pixels compared to the static algorithm by about 98 %.

In the future, we plan to support multiple labels per street, different anchor points, and different leader directions. Most importantly, we plan to combine our algorithm for placing billboards along the active route with our algorithm that embeds the labels of the remaining streets into these streets. The challenge will be to make sure that the combined algorithm is fast enough for real-time interaction. It would also be interesting to verify the findings of our survey (in which we used static figures) by means of a user study that provides interactive scenarios.

References

Eades, P. (1984). A heuristic for graph drawing. *Congressus Numerantium, 42*, 149–160.

Gemsa, A., Niedermann, B. & Nöllenburg, M. (2013). Trajectory-based dynamic map labeling. In L. Cai, S. W. Cheng & T. W. Lam (Eds.), *Proceedings of the 24th International Symposium on Algorithms and Computation* (ISAAC'13), Springer, LNCS, Vol. 8283, pp 413–423.

Imhof, E. (1975). Positioning names on maps. *The American Cartographer, 2*(2), 128–144.

Larson, K., van Dantzich, M., Czerwinski, M. & Robertson, G. (2000). Text in 3D: Some legibility results. In J. Begole (Ed.), *Proceedings of the 18th ACM Conference on Human Factors in Computing Systems* (*CHI'00*), pp 145–146.

Maass, S. & Döllner, J. (2006). Efficient view management for dynamic annotation placement in virtual landscapes. In A. Butz, B. Fischer, A. Krüger, P. Oliver (Eds.), *Proceedings of the 6th International Symposium on Smart Graphics* (*SG'06*), Springer, LNCS, Vol. 4073, pp 1–12.

Maass, S., Jobst, M., & Döllner, J. (2007). Depth cue of occlusion information as criterion for the quality of annotation placement in perspective views. In S. I. Fabrikant & M. Wachowicz (Eds.), *Proceedings of the 10th AGILE Conference,* pp. 473–486, Lecture Notes in GeoInformation and Cartography, Springer.

Schwartges, N., Wolff, A. & Haunert, J. H. (2014). Labeling streets in interactive maps using embedded labels. In *Proceedings of 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (*ACM-GIS'14*), pp. 517–520.

Vaaraniemi, M., Treib, M. & Westermann, R. (2012). Temporally coherent real-time labeling of dynamic scenes. In *Proceedings of 3rd International Conference on Computing for Geospatial Research Application* (*COM.Geo'12*), ACM, pp. 17:1–17:10.

Wigdor, D. & Balakrishnan, R. (2005). Empirical investigation into the effect of orientation on text readability in tabletop displays. In H. Gellersen et al. (Ed.) *Proceedings of the 9th European Conference on Computer Supported Cooperative Work* (*ECSCW'05*), Springer, pp 205–224.

Part IV
Improving Language and Representation in Spatial Computing

Aggregating Spatio-temporal Phenomena at Multiple Levels of Detail

Ricardo Almeida Silva, João Moura Pires, Maribel Yasmina Santos and Rui Leal

Abstract Spatio-temporal data are collected at high levels of detail (LoDs). Both spatial and temporal characteristics of data can be expressed at different LoDs. Depending on the phenomenon and the analytical goal, different LoDs can be suitable for a user's analysis since different LoDs may provide different perceptions of a phenomenon. It is crucial to model spatio-temporal phenomena having in mind that different LoDs can be useful in their analyses. We propose a granularities-based model in order to model spatio-temporal phenomena at multiple LoDs. It defines the concept of LoD and afterwards the atom generalization, granular synthesis and compressed granular syntheses set concepts to express a phenomenon at some LoD into a coarser one. This occurs in a semi-automatic way as the user just needs to define functions that create the compressed granular syntheses sets. A demonstration case was conducted applied to a real dataset about accidents in USA in which the model proposed proved to be useful to reduce the amount and complexity of data when the phenomenon is observed at coarser LoDs than the ones at which data is provided.

Keywords Granularity · Spatio-temporal data · Multiple levels of detail

R.A. Silva (✉) · J.M. Pires
NOVA-LINCS Lab, Universidade Nova de Lisboa, Lisbon, Portugal
e-mail: ricardofcsasilva@gmail.com

J.M. Pires
e-mail: jmp@di.fct.unl.pt

M.Y. Santos
ALGORITMI Research Centre, University of Minho, Braga, Portugal
e-mail: maribel.santos@algoritmi.uminho.pt

R. Leal
2GiveInsights, Lisbon, Portugal
e-mail: rui.pedro.leal@gmail.com

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_17

1 Introduction

Spatio-temporal data are being gathered at high levels of detail (LoDs) either from a spatial or temporal perspective, resulting into huge volumes of data to be processed, which can be further analyzed by users. Several examples can be found like spatio-temporal data collected from network sensors, remote sensing imagery; spatio-temporal data resulting from the usage of mobile devices (e.g., twitter); spatio-temporal data from monitoring sensors used in marine navigation (e.g., Automatic Identification System) or sensors embedded in vehicles (e.g., Tom Tom).

Spatio-temporal data embody spatio-temporal dynamism of natural phenomena or human activities. The spatio-temporal data attributes change over time or establish several relationships or interactions with the surrounding environment. The underlying space-time complexity makes the human ability to analyze and exploit spatio-temporal data (Keim et al. 2008) at such level(s) of detail unsuitable.

Looking at spatio-temporal phenomena, both spatial and temporal attributes of data can be expressed at different LoDs that can range, for instance, from grids with different cell sizes to cities or countries; from seconds to months or years. The LoD reflects the size of the units in which phenomena are observed and often aggregated/summarized (Andrienko et al. 2010). Consequently, the LoD that a phenomenon is being analyzed plays an important role in the analytical process. Depending on the phenomenon and the analytical goal, different LoDs can be suitable for a user's analysis once different LoDs will likely provide different perceptions of the same phenomenon. Therefore, it is crucial to model the spatio-temporal phenomena having in mind that different LoDs can be useful in the analysis of such phenomena and these can be related, for instance, through a spatial or temporal hierarchy.

In order to model spatio-temporal phenomena at multiple levels of detail, this paper extends our previous work (Pires et al. 2014). Our model lies on the granularity concept. Granularities create "lexicons" to describe phenomena. A granularities-based model allows us to model spatio-temporal phenomena at multiple LoDs. Each component of a phenomenon is described through granules of granularities. The extended granularities-based model provides the necessary formalism for defining the concept of LoD and afterwards the instruments to express a phenomenon at some LoD into a coarser one.

A demonstration case, applied to a real data set about accidents in USA, is presented in which the model proposed proved to be useful to reduce the amount and complexity of data when the phenomena are observed at coarser LoDs than the ones at which data is provided.

This paper is organized as follows. Section 2 presents related work about modelling spatio-temporal phenomena at different LoDs. In Sect. 3, the background needed about our previous work is given. Section 4 addresses the granularities-based model proposed in order to model spatio-temporal phenomena at multiple LoDs. Section 5 presents a demonstration case using the granularities-based model proposed. Section 6 concludes with some remarks about the undertaken work and some guidelines for future work.

2 Related Work

To model spatio-temporal phenomena at multiple LoDs, multiscale spatio-temporal models have been investigated, proposed by different researchers in different research areas like multirepresentation, multiresolution, multigranularity, and compressed data structures.

Multirepresentation aims at different viewpoints from a spatio-temporal phenomenon. Roughly, the approaches denoted by multirepresentation are based on extensions of the ER (Entity-Relationship) and UML (Unified Modeling Language) models. Several data models have been proposed in the literature in order to include spatial and temporal features in the database model including multiple representations (in some cases LoDs) as discussed in the survey presented by Parent et al. (2009). In their work, three requirements are also presented that should be verified in a multirepresentation model.

Firstly, the same object should be characterized through different sets of attributes or/and with different domain values. Secondly, a model should allow mapping one object to several objects or two different sets of objects. This is particularly useful when we change the spatial LoD, where objects may disappear and others may be grouped. Thirdly, a model should enable multiple representations of relationships. For instance, two regions might be modeled as spatially adjacent at lower scale but a more precise scale the regions are just near each other.

An example of a model holding the three requirements is MADS (Modeling Application Data with Spatio-temporal features) (Parent et al. 2006). Among the main drawbacks of multirepresentation is the fact that different LoDs, required by different applications, or the same application at different stages, can vary (Zhou et al. 2004). Bearing this in mind, the task of modeling a real-world phenomenon for which several spatial and/or temporal LoDs are needed can easily be very challenging. There are no pre-defined operations that take data from one spatial and/or temporal LoD to another one automatically, and everything needs to be defined by the user at the instances level.

The multiresolution is focused on the spatial component of the data. In an early work, Stell and Worboys (1998) define resolution or granularity as the level of discernibility between elements of a phenomenon that is being represented by the dataset. In general, multiresolution approaches hold data at the highest level of resolution and convert them to coarser ones, using known and pre-defined generalization operations (Stell and Worboys 1998; Weibel and Dutton 1999; Bertolotto and Egenhofer 2001). The generalization of spatial data is a non-trivial task and involves object simplification, which may lead to a change in the object geometry (e.g., a building can be represented by a polygon at a precise resolution, and by a point at a less precise resolution), dimensionality (e.g., at less precise resolutions, a building may be defined using less vertices than originally) and existence (e.g., eventually to represent that building is no longer relevant). More details about generalization operators can be found in Weibel and Dutton (1999).

Multiple resolutions of a phenomenon can be achieved through multiple representations, if we consider that the several representations concern the same geographical space and/or interval of time from the same perspective at different resolutions. To the best of our knowledge, the multiresolution approaches are focused on the map visualization (and the corresponding spatial generalization operators) and less with the computation of data at different LoDs.

Multigranularity approaches aim to hold data at several LoDs based on a granularity definition. There are several proposals for granularities definitions (Bettini et al. 2000; Camossi et al. 2006; Belussi et al. 2009; Pozzani and Zimányi 2012; Pires et al. 2014). Granularities can be related through relationships allowing one to compare and relate granules belonging to different granularities, which is useful to hold spatio-temporal data at different LoDs.

Camossi et al. (2006) propose a multigranularity approach to represent spatio-temporal information (vector approach) in object-oriented database management systems extending the ODMG standard. They define two new parametric data types. Spatial data types are defined through the $Spatial\langle G_s, \tau\rangle$ data type, where G_s is a spatial granularity (e.g., *Countries*, *Districts*) and τ being one of the ODMG types typically used to define conventional attributes like literal types (e.g., integer, float, etc.) or geometric types (like points, lines and polygons). Temporal or spatio-temporal data types are defined using the $Spatial\langle G_t, \tau\rangle$ data type where G_t is a temporal granularity (e.g., *Days*, *Months*) and γ can be any data type mentioned (including a spatial data type).

To $Spatial\langle G_s, \tau\rangle$ and $Temporal\langle G_t, \gamma\rangle$ data types, coarse and refinement functions can be assigned allowing to hold data at multigranularities (i.e., several LoDs). Coarse functions convert data from a granularity G_α to a coarser granularity G_β while refinement functions perform the opposite. There are coarse or refinement functions applicable to spatial geometrical attributes or spatial quantitative and temporal attributes. For example, coarse or refinement functions applied to spatial geometrical attributes can force some granules to modify their position and extent or to be merged, deleted, and splitted. Some coarse functions that can be applied on numerical types are: min, max, average. Using this approach, the user specifies, for each class attribute, what conversion functions can be used from a set of functions already defined.

The $Spatial\langle G_s, \tau\rangle$ data type indexes information of the type τ to spatial granules. Furthermore, the $Temporal\langle G_t, Spatial\langle G_s, \tau\rangle\rangle$ data type indexes the information of the type τ already indexed by spatial granules to temporal ones. Note that, when we define a temporal data type, the temporal granules are specifying the valid time of the information indexed to them. Another important aspect of this approach is that the indexed information will not be granules of some granularity but values of some type τ (belonging to some domain). As a result, in some scenarios, we cannot relate information at different LoDs. Consider the following class attributes: (i) $Spatial\langle G_{countries}, int\rangle$ storing information about the exact population number in each country; (ii) $Spatial\langle G_{countries}, String\rangle$ also storing information about the population number but with less precision so that the possible values are: (i) less

than one million (ii) one million or more and less than fifteen millions; (iii) fifteen or more millions. Although both variables refer to the same information, we cannot relate them by stating that the former is finer than the latter. This kind of reasoning is also important to relate spatio-temporal data at different LoDs.

More recently, a compressed hierarchical data structure was proposed in order to hold spatio-temporal data at multiple LoDs (Lins et al. 2013). They focus on providing real-time exploratory visualization for huge amounts of spatio-temporal events. The research of Lins et al. is aligned with the work here proposed. However, we aim to provide less detailed representations for spatio-temporal phenomena, in general, and not only for spatio-temporal phenomena stored by data sets of spatio-temporal events.

3 Granularity Theory

In our previous work, we present the foundations of a granularity theory devised to model spatio-temporal phenomena at different LoDs. This theory lies on two main concepts: granularity, and granularities-based model. Granularities perform divisions of a domain. Each division corresponds to a non-decomposable entity, mentioned as a granule. Through our definition, a granularity can be defined over any domain covering the definitions of temporal and spatial granularities proposed in the literature. A granularity was formally defined as follows.

Definition 1 (*Granularity*) Let $\mathcal{IS}$ be an index set; D be a domain; 2^{DS} the power set of the DS; and GS be a subset of the power set of the DS apart from the empty set $GS \subseteq 2^{DS} \backslash \{\emptyset\}$ such that any two elements are disjoint from each other. A granularity G is a bijective mapping:

$$G: GS \rightarrow \mathcal{IS} \tag{1}$$

A granularity G defines a division of a domain in a set of granules. A granule g_{ind} corresponds to a pair (g, ind) where $g \in GS$ and $ind \in \mathcal{IS}$. The extent of the granule g_{ind} is denoted by $E(g_{ind})$ which is g; the index value of the granule g_{ind} is denoted by $I(g_{ind})$ which corresponds to ind. The set of extents of granules is denoted by $GrS(G)$. The union of elements belonging to $GrS(G)$ defines the extent of a granularity $Ext(G)$.

Based on the presented granularity definition, spatial granularities like *States*, *Provinces* or *Countries*, and temporal granularities like *Days*, *Months* or *Years*, can be used to index facts adopting the granularities considered appropriate in a particular analytical context.

Granularities can be related through relationships allowing one to compare and relate granules belonging to different granularities, useful to hold spatio-temporal data at different LoDs. One commonly used relationship between granularities is the relationship *finer than*, i.e., a granularity G is *finer than* H if and only if each extent

of granule of G is contained in one extent of a granule of H ($G \preccurlyeq H$). For example, *County* is *finer than States*. More details about relations between granularities can be found in Pires et al. (2014). When a granularity G is *finer than* H then the extent of G is contained in the extent of H, i.e., $Ext(G) \subseteq Ext(H)$.

The granularities are instruments to create domains of discourse used to describe phenomena. Roughly, granularities that share the whole or part of their extension allow us to describe the same phenomenon using different LoDs. This happens when granularities are related by the *finer than* relationship. This characteristic of granularities is useful when a user needs to change the LoD in which a phenomenon is being described.

Granules by themselves are elements of the domain of discourse. We propose a granularities-based model that uses them to make statements about a phenomenon (Pires et al. 2014). A fundamental characteristic that makes our approach different from others is the fact that every feature of a phenomenon is described by a granule that refers to a granularity. Different statements can be related due to the relationships between granularities. Therefore, we can have different statements describing the same phenomenon at different LoDs, leading to a model that describes a phenomenon at multiple LoDs. The granularities-based model proposed does not formalize the instruments needed to express a phenomenon at some LoD into a coarser, in a semi-automatic way, which are the contributions of this work.

4 Extending the Granularities-Based Model

Our goal is to provide a model that allows us to look at a phenomenon at different LoDs. Bearing this in mind, an extension of the granularities-based model is proposed (Pires et al. 2014), providing the necessary formalism to define the concept of level of detail as well as the instruments to express a phenomenon at some LoD into a coarser one.

Throughout this section, we will illustrate our approach using a demonstration case based on data about accidents occurred in USA.[1] For each accident we consider the following information: *Space*, *Time*, *Victims*, where *Space* describes the location of the accident (in two-dimensional space), *Time* specifies the time when the accident occurred (in minutes), and *Victims* describes the number of injured individuals. For example: an accident occurred in a particular latitude and longitude on 1st January 2002 at 07:45 am causing one victim.

Let's start to define a granularities-based model for our example about accidents in USA which will be used to illustrate our approach as well as in the demonstration case (see Sect. 5). Firstly, a set of granularities is needed. The granularity *CoordsEight* is defined over the two-dimensional space where each granule represents a coordinate with eight decimal cases; similarly, consider the granularity

[1]Data are available at http://www.nhtsa.gov/FARS.

CoordsSeven. The granularity *Minutes* and *Hours* are defined over the time domain where each granule represents a minute and an hour, respectively. Lastly, consider the granularity *Numbers* defined over the natural numbers where each granule corresponds to an element of a corresponding domain. All these granularities define the set of granularities available in the granularities-based model of the accidents. In addition, $CoordsEight \preccurlyeq CoordsSeven$ and $\preccurlyeq Hours$.

In short, a granularities-based model is a set of atoms where each atom is a predicate symbol $P \in \mathcal{P}$ together with its arguments such that arguments are granules belonging to granularities $\mathcal{G}$. We start extending the granularities-based model by defining how to declare a predicate, and how to define a well-formed atom.

Let $P \in \mathcal{P}$ be n-ary predicate with a set of arguments denoted by $\text{Args}(P)$, and $\mathcal{G}$ a set of granularities. A predicate signature is of form $P\big(\{(\text{arg}, G_{(P,\text{arg})}) | \text{arg} \in \text{Args}(P) \wedge G_{(P,\text{arg})} \subseteq \mathcal{G}\}\big)$ that declares the set of valid granularities for each argument of P. An atom is of form $P(\tau)$ with $\tau = \{(\text{arg}, g) | \, \text{arg} \in \text{Args}(P) \wedge g$ is a granule of a valid granularity $G_{(P,\text{arg})}\}$. τ denotes the tuple of terms of an atom. For each argument one granule of one valid granularity is assigned.

Let's introduce the predicate symbol *accident* to define atoms that describe the occurrence of a single accident. For each argument of the predicate *accident* we declare the following set of valid granularities:

- $G_{(Accident,where)} = \{CoordsEight, CoordsSeven\}$
- $G_{(Accident,when)} = \{Minutes, Hours\}$
- $G_{(Accident,victims)} = \{Numbers\}$

A well-formed atom, of the *Accident* predicate, uses granules from the valid granularities declared for each argument. For example, o_1 = *accident*({*where*, $coordseight_1$), (*when*, 1-1-2002 07 h 45 a.m.), (*victims*, *one*)}) states that an accident occurred in the granule $coordseight_1$ on the granule 1-1-2002 07:45 a.m. causing *one* victim.

The set of granularities $\mathcal{G}$ together with the relation *finerthan*($\preccurlyeq$) define a partially ordered set (or poset) $\mathbb{G} = (\mathcal{G}, \preccurlyeq)$. Note that, the poset $\mathbb{G}$ may have a disconnected Hasse diagram, once the granularities belonging to $\mathcal{G}$ might be defined over disjoint domains, as happens in our example. The poset constituted by the granularities introduced together with the relation *finer than* leads to a Hasse diagram where the granularity *CoordsEight* is connected to *CoordsSeven*; *Minutes* is connected *Hours*; and *Numbers* isolated.

We assume that there is a *base granularity* for each set of valid granularities of each argument of a predicate P. A base granularity of an argument of a predicate P is a granularity that is related with any other granularity valid on such argument through the relation *finer than*. A base granularity in $G_{(P,\text{arg})}$ is formally defined as follows: $\exists! G_{base} \in G_{(P,\text{arg})} : G_{base} \preccurlyeq \text{G} \in G_{(P,\text{arg})}$. Looking at our example, the base granularity in $G_{(Accident,where)}$ is *CoordsEight*, the base granularity in $G_{(Accident,when)}$ is *Minutes* and in $G_{(Accident,victims)}$ is *Numbers*.

An atom describes something that happens in a spatio-temporal phenomenon. The set of granularities involved in an atom defines the LoD at which something is described. Let $\gamma = P(\{(\text{arg}_1, g_1), \ldots, (\text{arg}_n, g_n)\})$ be an atom; the set $\{(\text{arg}_1, G_1), \ldots, (\text{arg}_n, G_n)\}$ describing the granularity used on each argument defines its level of detail $LoD(\gamma)$. The $LoD(o_1)$ is: $\{(where, CoordsEight), (when, Minutes), (victims, Numbers)\}$. We define the valid LoDs of a predicate as follows.

Definition 2 (*Valid Levels of Detail of a Predicate*) Let $P \in \mathcal{P}$ be n-ary predicate and its signature $P\left(\left\{\left(\text{arg}, G_{(P,\text{arg})}\right) | \text{arg} \in \text{Args}(P) \wedge G_{(P,\text{arg})} \subseteq \mathcal{G}\right\}\right)$ defining a set of valid granularities for each argument; then $\mathcal{L}^P = \bigotimes_{\text{arg} \in \text{Args}(P)} G_{(P,\text{arg})}$ is the set of valid LoDs of the predicate P.

Two valid levels of detail α and β of a predicate P can be related based on the granularity relationship *finer than*. We introduce the *more detailed than* relation between LoDs.

Definition 3 (α *is more detailed than* β) Let $P \in \mathcal{P}$ be n-ary predicate; let $\alpha \in \mathcal{L}^P$ and $\beta \in \mathcal{L}^P$ be two valid LoDs of P such that $\alpha = \{(\text{arg}_1, G_1), \ldots, (\text{arg}_n, G_n)\}$ and $\beta = \{(\text{arg}_1, H_1), \ldots, (\text{arg}_n, H_n)\}$; α is more detailed than β, $\alpha \preccurlyeq_L \beta$, if and only if, $G_i \preccurlyeq H_i$ for all $1 \leq i \leq n$.

The set of all valid levels of detail $\mathcal{L}^P$ of a predicate $P \in \mathcal{P}$ with the relation *is more detailed than* ($\preccurlyeq_L$) define a poset: $\mathbb{L}^P = (\mathcal{L}^P, \preccurlyeq_L)$. There is only one least level of detail α in $\mathbb{L}^P$ such that for every level of detail β in $\mathbb{L}^P$, $\alpha \preccurlyeq_L \beta$. Note that, the least level of detail of a predicate P is composed by the set of base granularities of the corresponding arguments, which we denote by the base level of detail of P.

In our example, the Hasse diagram for the poset $\mathbb{L}^{Accident}$ is illustrated in Fig. 1, in which the base LoD of the *Accident* predicate corresponds to the LoD $\alpha = \{(where, CoordsEight), (when, \{Minutes\}), (victims, Numbers)\}$.

In order to have atoms at multiple levels of detail, we propose to take an atom in one LoD and express it at a coarser one. Let's introduce the atom generalization concept.

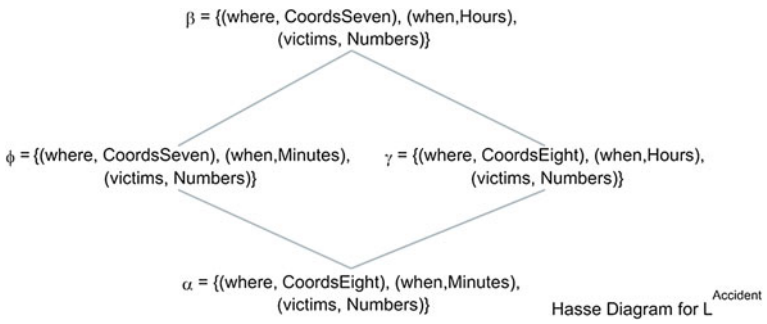


Fig. 1 The Hasse diagram for the poset $\mathbb{L}^{Accident}$

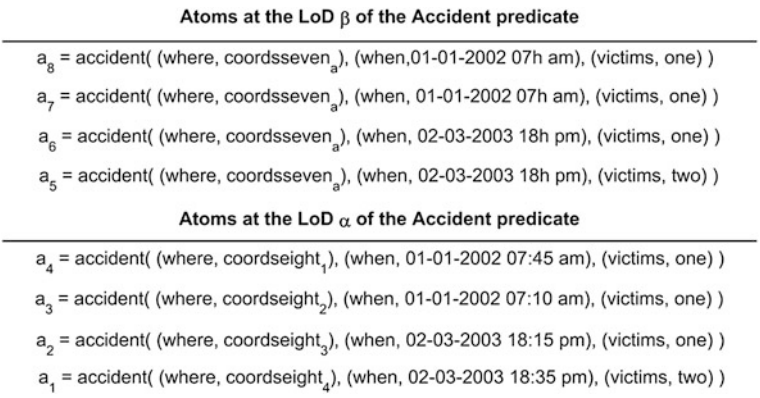


Fig. 2 Example of atoms at different valid LoDs of the Accident predicate

Definition 4 (*Atom Generalization*) Let $P \in \mathcal{P}$ be n-ary predicate with a set of arguments $\arg_1, \ldots, \arg_n$; let a_α be an atom at a valid level of detail α of P; let β be a valid level of detail of P such that $\alpha \preccurlyeq_L \beta$; then, a function Gen: $(a_\alpha, \beta) \rightarrow a_\beta$ express the atom $a_\alpha = P((\arg_1, g_1), \ldots, (\arg_n, g_n))$ at the level of detail α in an atom $a_\beta = P((\arg_1, h_1), \ldots, (\arg_n, h_n))$ at the level of detail β such that $E(g_i) \subseteq E(h_i)$ for all $1 \leq i \leq n$.

Let's consider the atoms $a_1, \ldots, a_4$, shown in Fig. 2, expressed at the base level of detail α of the *Accident* predicate. The atoms at such LoD are describing the location of the accident with eight decimal cases and when it occurred with a precision of minutes. However, the exact minute that an accident occurs can be irrelevant. Also, the description of the location with a precision of eight decimal cases may also be too detailed. This way, consider that it would be desirable to describe the location through seven decimal cases.

Using the atom generalization, we can produce a set of atoms at the valid LoD β based on the set of atoms at the base LoD α of the *Accident* predicate. For the sake of simplification, we are not making the actual coordinates available. The atom a_4 can be generalized into the atom a_8 once the extent of the granule $coordseight_1$ is contained by the extent of the granule $coordsseven_a$: $E(coordseight_1) \subseteq E(coordsseven_a)$. The same relation exists between the granules used in the other arguments: E(01-01-2002 07:45 a.m.) $\subseteq$ E(01-01-2002 07 h a.m.); and $E(one) \subseteq E(one)$. Similarly, the atom generalization is applied for the remaining atoms at the base level of detail α.

The atom generalization provides an instrument to look at a phenomenon at multiple LoDs. However, it is not enough to reduce the volume and complexity of a phenomenon's representation. A granularities-based model may contain equal atoms, i.e., atoms defined by the same tuple of terms in some valid LoD of a predicate P in spite of the fact that these are referring to different events that occurred in a phenomenon.

As can be seen in Fig. 2, the atoms at the valid LoD α of the *Accident* predicate are discernable from each other while at the valid LoD β some atoms are equal, namely a_7, a_8. Note that, they are describing distinct accidents.

In general, at a valid LoD of a predicate P, there may be atoms equal to each other. Furthermore, as the atoms are described through coarser valid LoDs of P, the number of equal atoms tends to increase. When there are equal atoms at some LoD of a predicate P, we are interested in performing synthesis of atoms in order to reduce the number of atoms that describe a phenomenon. Thereby, we introduce the concept of granular synthesis.

Beforehand, and without loss of generality, we assume that any atom of form $P((\arg_1, g_1), \ldots, (\arg_n, g_n))$ can be expressed equivalently as $G_{Syn}(P((arg_1, g_1), \ldots, (arg_n, g_n)), 1)$, where G_{Syn} is a reserved predicate such that the first argument contains an atom of a predicate P and the second one indicates the number of occurrences of such atoms which, in this case, is one.

Definition 5 (*Granular Synthesis*) Let $P \in \mathcal{P}$ be n-ary predicate; let τ be a tuple of terms; let $\mathbb{A}$ be a set of atoms at a valid LoD of P such that any atom $a \in \mathbb{A}$ is of form $G_{Syn}(P(\tau), c)$; then, a function $f: \mathbb{A} \rightarrow G_{Syn}(P(\tau), c)$ produces a granular synthesis where c is the sum of all counts c in $\mathbb{A}$ such that $c \in \mathbb{N}$.

A granular synthesis makes a summary of a set of equal atoms at a valid level of detail of a predicate P. Thus, a granular synthesis is an instrument to reduce the volume of atoms at some LoD of a predicate P. As shown in Fig. 3, the atoms a_7, a_8 resulted in the granular synthesis a_9. The remaining atoms of the *Accident* predicate are expressed also as granular syntheses in spite of their count being equal to one.

When we take a phenomenon at some LoD in order to express it into a coarser LoD, we intend to allow some loss of information as a way to achieve more meaningful statements. This is related with the Law of Incompatibility stated by Zadeh (1965), "*As complexity rises, precise statements lose meaning and meaningful statements lose precision*".

The set of atoms (or granular syntheses) at valid LoD of a predicate P may be expressed in a more compact way if we are open to lose some information. Some

Atoms at the LoD β of Accident (expressed through granular syntheses)
$a_9 = G_{Syn}$(accident((where, coordsseven$_a$), (when, 01-01-2002 07h am), (victims, one)), 2)
$a_{10} = G_{Syn}$(accident((where, coordsseven$_a$), (when, 02-03-2003 18h pm), (victims, one)), 1)
$a_{11} = G_{Syn}$(accident((where, coordsseven$_a$), (when, 02-03-2003 18h pm), (victims, two)), 1)
Atoms at the LoD β of Accident
a_8 = accident((where, coordsseven$_a$), (when, 01-01-2002 07h am), (victims, one))
a_7 = accident((where, coordsseven$_a$), (when, 01-01-2002 07h am), (victims, one))
a_6 = accident((where, coordsseven$_a$), (when, 02-03-2003 18h pm), (victims, one))
a_5 = accident((where, coordsseven$_a$), (when, 02-03-2003 18h pm), (victims, two))

Fig. 3 Example of granular syntheses at the LoD **β** of the Accident predicate

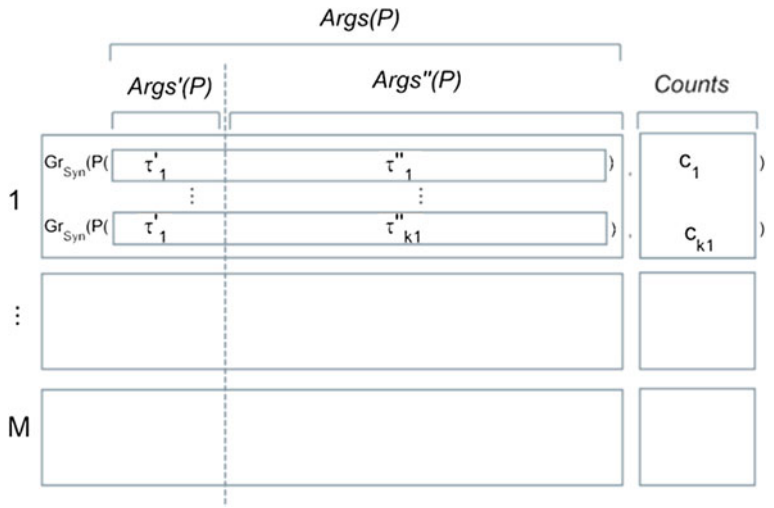


Fig. 4 A diagram of partitioning a set of granular syntheses at a LoD of a predicate P

atoms may share a subset of terms of the tuple of terms in P. For example, in Fig. 4, the granular syntheses a_9, a_{10}, a_{11} share the subset of terms: $\{(where,\ coordseven_a)\}$ in the *Accident* predicate. For those scenarios, we propose to compress a set of granular syntheses.

As shown in Fig. 4, a n-ary predicate $P \in \mathcal{P}$ has a set of arguments $\text{Args}(P)$, which can be split as the set of arguments to hold $\text{Args}''(P) \subset \text{Args}(P)$ and the set of arguments to compress $\text{Args}''(P) \subset \text{Args}(P)$ such that $(\text{Args}'(P) \cap \text{Args}''(P)) = \emptyset \wedge (\text{Args}'(P) \cup \text{Args}''(P)) = \text{Args}(P)$. Thereby, a tuple of terms of an atom of a predicate P can be seen as $\tau = \tau' \cup \tau''$ such that $\tau' = \{(\arg, g) | (\arg, g) \in \tau' \wedge \arg \in \text{Args}'(P)\}$ and $\tau'' = \{(\arg, g) | (\arg, g) \in \tau'' \wedge \arg \in \text{Args}''(P)\}$.

Consider $\mathbb{GS}(P, \alpha)$ a set of granular syntheses of a predicate P at a LoD α. Given a subset of arguments of P, $\text{Args}'(P)$, we can partition $\mathbb{GS}(P, \alpha)$, denoted by $\mathbb{GS}(P, \alpha, \text{Args}'(P)) = \{\mathbb{K}_1, \mathbb{K}_2, \ldots, \mathbb{K}_m\}$ such that all elements in $\mathbb{K}_m$ share the same τ'_m. In Fig. 4, the first subset of the partition K_1 has $k1$ granular syntheses wherein all tuple of terms τ''_{k1} are different while all tuple of terms τ'_1 are equal.

We propose to compress each subset $\mathbb{K}_m$ of the partition as shown in Fig. 5. Besides the τ'_m which is constant for all granular syntheses in $\mathbb{K}_m$, the τ'' are all different from each other. A compressed summary can be achieved in different ways. It may be just counting the number of granular syntheses; statistical measures over the counts of the set of granular syntheses like sum, max; or compression operations based on all tuple of terms τ''; or even a combination of several compression operations. Let's introduce the concept of compressed granular syntheses set.

Definition 6 (*Compressed Granular Syntheses Set*) Let $P \in \mathcal{P}$ be n-ary predicate; let $\mathbb{K}_i$ be a partition in a valid LoD of a predicate P $(1 \leq i \leq m)$; then, a declared function $g: \mathbb{K}_i \rightarrow \mathbb{C}_{Syn}(P(\tau'_1), compressedsummary)$ produces a compressed summary.

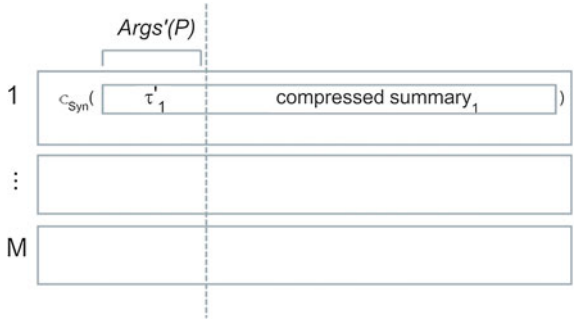


Fig. 5 A compressed representation of a set of granular syntheses

A compressed granular syntheses set of granular syntheses at the LoD β of Accident

$a_{12} = G_{Syn}$(accident((where, coordsseven$_a$)), (count, 3), (sum, 4))

Granular Syntheses at the LoD β of Accident

$a_{11} = G_{Syn}$(accident((where, coordsseven$_a$), (when, 02-03-2003 18h pm), (victims, two)), 1)

$a_{10} = G_{Syn}$(accident((where, coordsseven$_a$), (when, 02-03-2003 18h pm), (victims, one)), 1)

$a_9 = G_{Syn}$(accident((where, coordsseven$_a$), (when, 01-01-2002 07h pm), (victims, one)), 2)

Fig. 6 A compressed granular synthesis set at the LoD **β** of the Accident predicate

Looking at our example in Fig. 6, based on the term $\{(where,\ coordseven_a)\}$, we can produce the compressed granular syntheses set a_{12} which is the result of applying a function to compress the granular syntheses a_9, a_{10}, a_{11}. The function used counts the number of granular syntheses and sums the victims. The compressed granular synthesis set a_{12} provides a synthesis about the occurrence of accidents in the granule $coordseven_a$ in spite of the loss of information. This kind of reasoning can be useful in many analytical contexts.

The concepts introduced and illustrated lead to a model that allows us to look at a phenomenon and analyze it at different levels of detail, formalized as follows.

Definition 7 (*Granularities-based Model*) Let $\mathcal{G} = \{\mathcal{A}(G_1), \ldots, \mathcal{A}(G_n)\}$ be a set of annotated granularities, and $\mathcal{P}$ a set of predicates. Each predicate has defined its signature. A granularities-based model $\mathcal{M}$ is a set of well-formed atoms.

- $P(\tau)$ with $\tau = \{(\text{arg}, g) | \text{arg} \in \text{Args}(P) \wedge g$ is a granule of a valid granularity $G_{(P,\text{arg})}\}$.
- $G_{Syn}(P(\tau), \text{c})$ such that $c \in \mathbb{N}$.
- $\mathbb{C}_{Syn}(P(\tau'), compressedsummary)$ with $\tau' = \{(\text{arg}, g) | \text{arg} \in \text{Args}'(P) \subset \text{Args}(P) \wedge g$ is a granule of a valid granularity $G_{(P,\text{arg})}\}$.

Regarding the granularities-model proposed, the user needs to define the set of granularities, the predicates and a set of functions that can be used to create the compressed granular syntheses sets. Based on the predicates, the defined base granularities and the data describing a spatio-temporal phenomenon, the model can be filled with the set of atoms at the base predicates LoDs.

Each predicate provides a representation of the phenomenon like the multirepresentation approaches, but unlike them there is no need to define everything at the instances level. Once the atoms at the base LoD of the predicates are produced, the phenomenon can be expressed into other coarser LoDs in a semi-automatic way, as the user just needs to define the functions that create the compressed granular syntheses sets. Unlike the multiresolution approaches, the granularities-based model can express a phenomenon in several LoDs, and not just in several spatial LoDs. Last but not least, unlike the multigranularity approaches we provide instruments to create granular syntheses (as well as compressed granular syntheses set) and not just a way of converting information from one granularity to another.

5 Demonstration Case

To conduct the demonstration case, a prototype was developed that follows the client-server architecture for the current implementation of the granularities-based model. The server was implemented in Java and it is responsible for listening to client requests, processing them and retrieving the appropriate results. The browser-based client handles user interaction and data presentation and is written in Javascript, HTML5, WebGL, and it uses Leaflet to support the visualization of data on a map.

A demonstration case is based on data about accidents in USA (see Sect. 4). The dataset contain the accidents occurred between 2001 and 2013, which corresponds to 450.710 geo-referenced accidents.

A model $\mathcal{M}_{Accidents}$ is defined based on the *Accident* predicate. The valid granularities for each argument were also defined as displayed in Fig. 7. In our prototype, the Leaflet uses the Google Map layer as background in which we display data. We adopt a set of spatial granularities, one for each map zoom level, defined over bi-dimensional space such that $One \preccurlyeq Two \preccurlyeq \cdots \preccurlyeq Tweenty{-}two$. These are valid for the argument *where*. For the argument *when*, the granularities were defined over the time domain and are related in the following way: $Minutes \preccurlyeq Hours \preccurlyeq \text{Days} \preccurlyeq Weeks$. Furthermore, one function that sums the number of accidents and sums the number of victims is defined in order to create compressed granular syntheses sets.

We are interested in having a glimpse of the spatial distribution of accidents occurred. Thus, the set of atoms at the base LoD of the *Accident* predicate (identified by 1 in the LoD axes in Fig. 8) were generalized for the valid LoD of the *Accident* predicate: $2 = \{(where, Two), (when, \{Minutes\}), (victims, Numbers)\}$; $3 = \{(where, Three), (when, \{Minutes\}), (victims, Numbers)\}$ and so on. Note that,

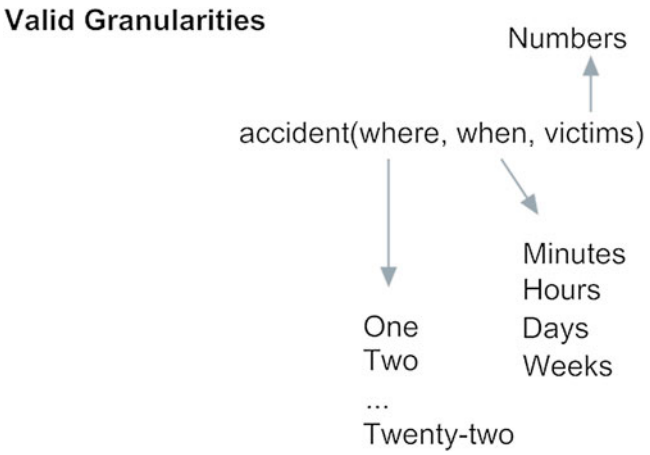


Fig. 7 The valid granularities regarding the Accident predicate

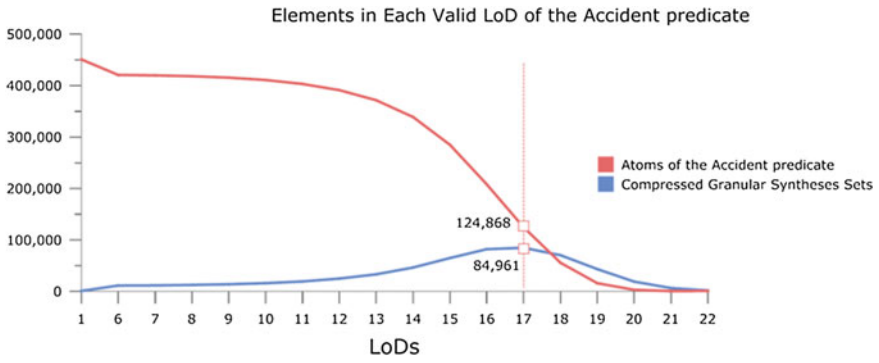


Fig. 8 The number of atoms, compressed granular syntheses sets concerning the Accident predicate by LoD

the granular syntheses of the *Accident* predicate are in form $G_{Syn}(Accident(\tau), c)$. Recall that, to produce compressed granular syntheses sets we need to indicate the arguments to hold and the function that produced the compressed summary. In this case, the argument to hold corresponds to *where* and the function already defined sums the counts c and the number of victims for each granule in the *where* argument, producing compressed summaries.

For each LoD, the number of atoms of the *Accident* predicate which were not compressed (or in others words granular syntheses where the c value is equal to one) and the number of compressed granular syntheses sets are displayed in Fig. 8. As we move to coarser LoDs, the number of the atoms decreases due to the computation of compressed granular syntheses sets. The reduction of atoms for each LoD is shown in Fig. 9. The reduction corresponds to one less the sum of the

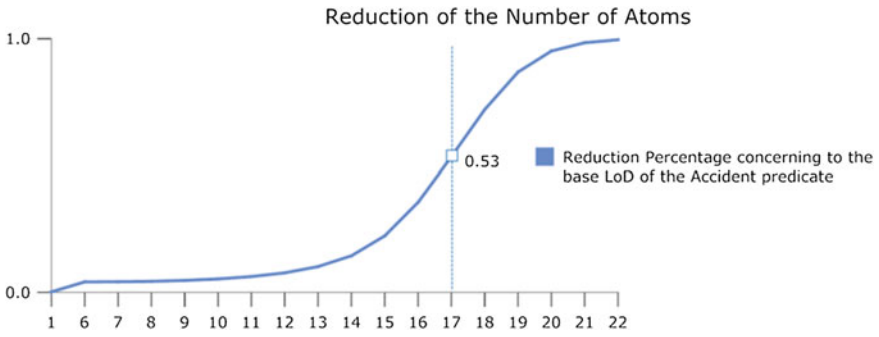


Fig. 9 The reduction percentage of atoms concerning the Accident predicate by LoD

number compressed granular syntheses sets and the number of atoms of the *Accident* predicate divided by the total atoms at base LoD *P* which is 450.710.

Figure 10 displays two maps in the zoom level four of the map. The first one displays the set of atoms at the base LoD of the *Accident* predicate where each atom's representation consist is a red "point" with 0.3 value of transparency (being 0 completely transparent and 1 opaque). The second map displays the compressed granular syntheses sets and the atoms that exist at the LoD 17 concerning the *Accident* predicate. Here, the transparency of the compressed granular syntheses sets is given by the value that holds the sum of the number of accidents: (i) for values between 1 and 3, 0.5 was assigned; (ii) for values greater than 3, 0.8 was assigned.

Looking at Fig. 10, the perception about the spatial distribution of accidents is not affected in the LoD 17, at map zoom level four, in spite of less 53 % of items than at base LoD of *Accident* predicate being displayed. In spite of the decrease in precision, the user keeps the same analytical capability.

Moreover, the set of atoms at the base LoD of the *Accident* predicate were generalized for the following valid LoDs of the *Accident* predicate:

- $M = \{(where, one), (when, \{Minutes\}), (victims, Numbers)\}$
- $M = \{(where, one), (when, \{\ Hours\}\ ), (victims, Numbers)\}$
- $\mathrm{D} = \{(where, one), (when, \{Days\}), (victims, Numbers)\}$
- $\mathrm{W} = \{(where, one), (when, \{Weeks\}), (victims, Numbers)\}$

Figure 11 displays the total number of accidents that occurred by minute (LoD_M), by hour (LoD_H), by day (LoD_D) and by week (LoD_W). The reduction of the number of atoms in the LoD_M, LoD_H, LoD_D, LoD_W corresponds to approximately 6, 75, 98, 99 %, respectively. According to the LoD adopted our understanding of the phenomenon is affected. In fact, in the LoD_W, it becomes clear the cyclic pattern concerning the occurrence of accidents. This reinforces the idea that the proper LoD should be chosen according to the phenomenon and analytical goal.

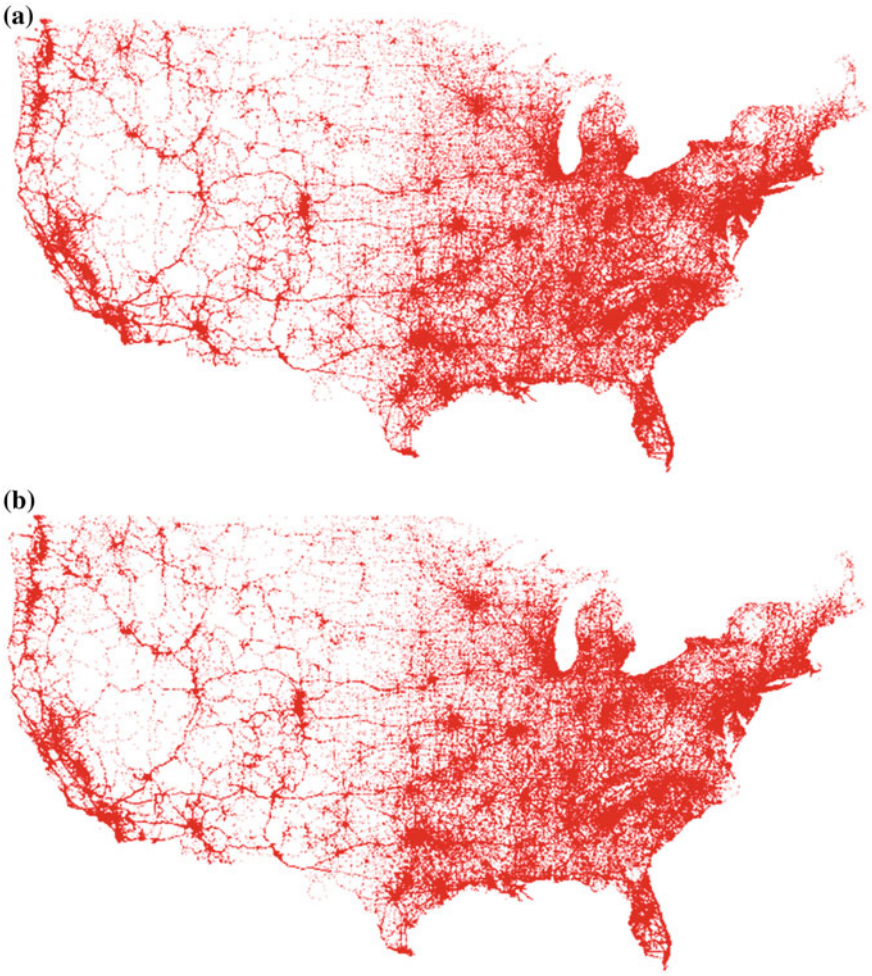


Fig. 10 A glimpse at the location of the accidents: **a** data at the base LoD of the Accident predicate; **b** data at the LoD 17 of the Accident predicate

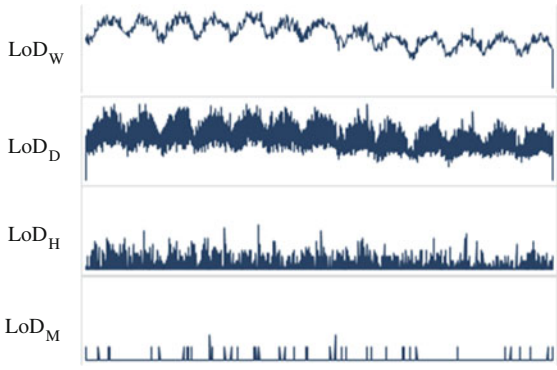


Fig. 11 Data at the LoD_M, LoD_H, LoD_D, and LoD_W

6 Conclusions and Future Work

This paper presents a model that allows us to look at spatio-temporal phenomena at multiple LoDs. The granularities-based model defines the concept of LoD and afterwards the atom generalization, granular synthesis and compressed granular syntheses set concepts to express a phenomenon at some LoD into a coarser. Once the atoms at the base LoD of the predicates are produced, the phenomenon can be expressed into other coarser LoDs in a semi-automatic manner, as the user just needs to define the functions that create the compressed granular syntheses sets.

A demonstration case, applied to a real data set about accidents in USA, is presented in which the model proposed proved to be useful to reduce the amount and complexity of data when the phenomenon are observed at coarser LoDs than the ones at which the data is provided.

Nevertheless, the development of the model is not finished. We intend to drop the assumption that for each argument of a predicate only one granule can be assigned. This way, we will be able, for example, to hold more complex spatial objects like lines and polygons by describing them through a sequence of granules belonging to a certain spatial granularity. As a result, the atom generalization will be responsible for changing the dimensionality and object geometry in some contexts when expressing an atom (with such spatial objects) from one LoD into a coarser one. Also related with the atom generalization, we intend to define an automatic and configurable instrument to decide whether an atom should exist or not at some LoD. In other words, we intend to incorporate the *Generalization-reduction-disappearance* process (Laurini 2014) in the granularities-based model. Moreover, future work can be directed to further experimentation of the model proposed.

References

Andrienko, G., et al. (2010). Space, time and visual analytics. *International Journal of Geographical Information Science, 24*(10), 1577–1600.

Belussi, A., Combi, C. & Pozzani, G., 2009. Formal and conceptual modeling of spatio-temporal granularities. In *Proceedings of the 2009 International Database Engineering & Applications Symposium on—IDEAS '09* (p. 275). New York, USA: ACM Press.

Bertolotto, M., & Egenhofer, M. J. (2001). Progressive transmission of vector map data over the world wide web. *GeoInformatica, 5*(4), 345–373.

Bettini, C., Jajodia, S., & Wang, S. (2000). *Time granularities in databases, data mining, and temporal reasoning*. Berlin: Springer.

Camossi, E., Bertolotto, M., & Bertino, E. (2006). A multigranular object-oriented framework supporting spatio-temporal granularity conversions. *International Journal of Geographical Information Science, 20*(5), 511–534.

Keim, D., et al. (2008). Visual analytics: Definition, process, and challenges. In A. Kerren et al. (Eds.), *Information Visualization*. Lecture Notes in Computer Science (pp. 154–175). Berlin, Heidelberg: Springer.

Laurini, R. (2014). A conceptual framework for geographic knowledge engineering. *Journal of Visual Languages & Computing, 25*(1), 2–19.

Lins, L., Klosowski, J. T., & Scheidegger, C. (2013). Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics, 19*(12), 2456–2465.

Parent, C., Spaccapietra, S., & Zimányi, E. (2006). The MurMur project: Modeling and querying multi-representation spatio-temporal databases. *Information Systems, 31*(8), 733–769.

Parent, C., et al. (2009). Multiple representation modeling. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of database systems (*pp. 1844–1849). Berlin: Springer.

Pires, J. M., Silva, R. A., & Santos, M. Y. (2014). Reasoning about space and time: Moving towards a theory of granularities. In *Computational Science and Its Applications–ICCSA 2014* (pp. 328–343). Berlin: Springer.

Pozzani, G., & Zimányi, E. (2012). Defining spatio-temporal granularities for raster data. In *Data Security and Security Data* (pp. 96–107). Berlin: Springer.

Stell, J., & Worboys, M. (1998). Stratified map spaces: A formal basis for multi-resolution spatial databases. In *Proceedings 8th International Symposium on Spatial Data Handling*. Department of Computer Science, Keele University, Staffordshire, UK ST5 5BG (pp. 180–189).

Weibel, R., & Dutton, G. (1999). Generalising spatial data and dealing with multiple representations. *Geographical information systems, 1*, 125–155.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control, 8*(3), 338–353.

Zhou, X., et al. (2004). Multiresolution spatial databases: Making web-based spatial applications faster. In J. Yu et al. (Eds.), *Advanced web technologies and applications SE—5*. Lecture Notes in Computer Science (pp. 36–47). Berlin, Heidelberg: Springer.

Designing a Language for Spatial Computing

Werner Kuhn and Andrea Ballatore

> *"One of the main reasons why software projects fail is the lack of communication between the business users, who actually know the problem domain, and the developers who design and implement the software model."*
>
> (Ghosh 2011).

Abstract We present the design rationale underlying a language for spatial computing and sketch a prototypical implementation in Python. The goal of this work is to provide a high-level language for spatial computing that is executable on existing commercial and open source spatial computing platforms, particularly Geographic Information Systems (GIS). The key idea of the approach is to target an abstraction level higher than that of GIS commands and data formats, yet meaningful within and across application domains. The paper describes the underlying theory of spatial information and shows its evolving formal specification. An embedding in Python exemplifies access to commonly available implementations of spatial computations.

Keywords Spatial computing · Domain-specific language · Core concepts

1 Introduction

If spatial computing is to realize its often-proclaimed potential across disciplines, geographic information science needs to convey a clearer picture of what GIS and related technologies are good for. This picture must focus on information *contents*

W. Kuhn (✉)
Center for Spatial Studies, Department of Geography, University of California, Santa Barbara, USA
e-mail: werner@ucsb.edu

A. Ballatore
Center for Spatial Studies, University of California, Santa Barbara, USA
e-mail: aballatore@spatial.ucsb.edu

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_18

and user *questions*, rather than on data formats (e.g., raster and vector) and system commands, which dominate the current image of GIS. It should be a value proposition that is meaningful across application domains, while avoiding overgeneralizations (such as reducing all spatial information to a single form) or obscure terminology (such as abstract ontological terms).

Existing abstractions like the geo-atom (Goodchild et al. 2007) and the notion of generalized fields (Camara et al. 2014) are helpful in generating such a picture, but they attempt to squeeze all spatial information into a single form, avoiding content distinctions. Similarly, spatial analysis reduces spatial information essentially to products of random point processes. In the absence of a clear but nuanced picture of spatial information and computing, many potential users continue to believe that GIS is primarily used to make and store maps.

The problem of software requiring users to speak a language they may not be familiar with is as old as computing itself. In the early 1980s, computer scientists and psychologists started to describe and address it systematically. Don Norman coined the terms "Gulf of Execution" and "Gulf of Evaluation" (Norman 1986) to describe the gap between how users think and how computers require them to talk (execution) as well as how they talk back to users (evaluation). By the early 1990s, usability engineering had become a much-researched part of system design and implementation, generally and in the GIS area (Egenhofer and Kuhn 1999). The user interfaces of GIS rapidly changed from command line to direct manipulation interfaces, including visual programming languages (VPL).[1]

Yet, this re-organization of GIS languages has largely remained syntactic. The semantic questions, concerning what contents users want to talk *about* at a GIS interface, have hardly been addressed. Consequently, there is still no commonly accepted classification of the types of spatial or geographic information (as opposed to the data types) that are handled by GIS. Organizing GIS commands bottom-up turned out to be too hard (Albrecht 1998), and ontologists have not yet come up with a top-down structure capturing what is special about spatial information. Geospatial ontologies, on the other hand, specify application domains like hydrology or land cover, without attempting the generalization sought here. The prolonged absence of an answer to what spatial or geographic information is about is further aggravated by *pride in complexity*: researchers and practitioners mastering GIS all too often believe that what they have spent so much effort (and money) on learning is inherently complex and needs to stay that way.

Standardization, meanwhile, has quietly abandoned its original aspiration of defining service interfaces for spatial computing at the content level. In the absence of theories to inform the design of such interfaces, OGC and ISO had to pursue a modern (service-oriented) form of data exchange, based on GML. The idea of a software-independent essential model of geospatial computing fell by the wayside, and the language of geographic information standards became one of software

[1]Industrial products include *ModelBuilder* in ArcGIS and the *Workflow Designer* in Autodesk Map 3D.

technologies, rather than of information contents. Sadly, this is mainly a result of researchers keeping themselves busy developing and testing industrial software specifications, rather than developing theories of spatial information and computing.

Thus, a quarter century into GIScience research and standardization, our field still fails to communicate what GI and GIS are about. Evidence for this can be found in the heterogeneous organization of textbooks and curricula. As a result, teaching GIS tends to default to software training, and attempts to explain the potential of GIS to, say, an epidemiologist, an economist, or a historian, tend to leave these colleagues with an impression that they better leave GIS to specialists.

Contrast this situation with that in statistics. The two fields play similar cross-disciplinary roles, but statistics has had time and intellectual aspiration to achieve a mature self-image and understanding of its tools. Simplifying somewhat (as one needs to, for this purpose), the field of statistics comes with a set of core concepts, ranging from random variables through distributions and correlations to statistical tests. If one wants to do statistics, these concepts provide access to and transfer between a large and growing variety of statistical computing tools, all of them more or less understanding and speaking these terms.

This paper presents a novel approach to the problem of describing spatial information and computations above the level of GIS data formats and commands. It proposes to use the previously defined *core concepts of spatial information* (Kuhn 2012) as a vocabulary for a language of spatial computing, supplying a high-level view of operations on these concepts. We first discuss what domain the language should be for (Sect. 2), then recall the core concepts (Sect. 3) and specify them through their operations (Sect. 4), preparing for a sketch of an ongoing implementation in Python (Sect. 5). Conclusions and a discussion of future work wrap up the paper (Sect. 6).

2 What Domain?

While it remains to be seen whether "geographic" or "spatial" is the better scope for a language of spatial computing, our work rests on the assumption that most computational techniques developed for geographic applications are applicable to other spaces as well. For example, fields or networks are both useful structures across many scales, dimensions, and applications. Therefore, we refrain from any attempts to limit the application domains, and thus the scope of the domain-specific language, to geographic or other spaces. Still, the spaces considered here primarily are those of human experience in one to three dimensions. The applications are primarily geographic (dealing with neighborhoods or river catchments, for example) or indoor environments (dealing with floors, rooms or hallways), but other spaces remain in scope. We refer to this broad scope as our domain and to areas like ecology or economics as applications. Figure 1 provides an overview of the conceptual and technology layers involved in our proposal.

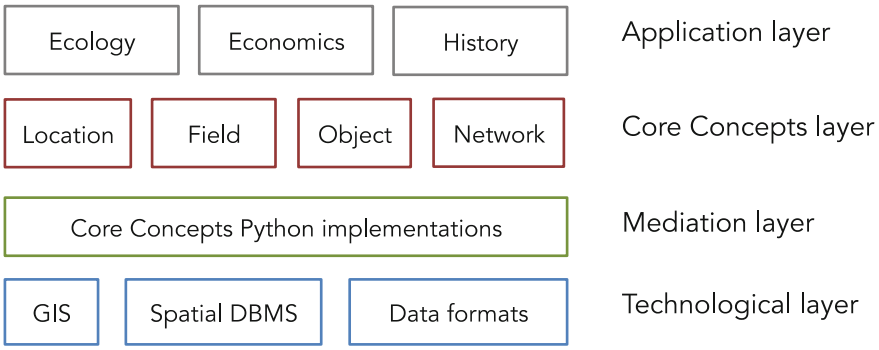


Fig. 1 Overview of the spatial computing domain

Originally, we planned to map the core concepts to domain specializations (for example, translating the core concept of network to the domain specialization of road network), but we found it beneficial to preserve the language's generality and abstraction even within applications. A domain mapping remains a valid option for any specialized domain language in which conceptualizations can be fixed. However, it is a characteristic of spatial computing that different conceptualizations co-exist, for example, of roads as networks or complex objects. Thus, we envisage a scenario where domain practitioners express their spatial questions in the proposed high-level language, specializing their entities when necessary, and operating at a cross-domain conceptual level.

3 Core Concepts of Spatial Information

This section summarizes the core concepts of spatial information in their latest form of seven concepts—see also (Kuhn 2012). These connect spatial thinking to computing and provide a high-level vocabulary with which to ask and answer questions about phenomena in space and time. Spatial information is spatiotemporal by default in today's practice, so that the concepts do not distinguish spatiotemporal from spatial information.

A remark on the underlying notion of *information* is in order. Information answers questions. It exists only in the minds of people and not in computers, nor in the world outside of human minds or computers. Spatial information gets generated by humans observing the world with a reference to location and extent. Concepts of spatial information are the elements from which humans build mental representations of the world. Computers hold *data* about the world, recording human or technical observations and used to answer questions. It is often convenient to use one and the same term for these three aspects of information. For example, the term *object* can be used to refer to something physical in the world (say, a house), its

Table 1 Overview of the core concepts of spatial information

Core content concepts					Core quality concepts
Location	Field	Object	Network	Event	Granularity
					Accuracy

mental representation (your idea of that house), and the data describing it (in a cadastral database).

The seven core concepts of spatial information comprise five concepts of information *content* and two concepts of information *quality*. The latter can be considered meta-information concepts, as they apply to all content concepts. Table 1 shows an overview.

This small set of core concepts of spatial information inevitably misses some ideas or relegates them to sub-concepts or non-core status. Other views of what is "core" can be found in (Golledge 1995) or (Janelle and Goodchild 2011). Picking core concepts is mainly a terminological and granularity choice with which one decides on how to talk about spatial information. Their suitability needs to be judged through applications, for the purpose of which we are designing the domain language around them.

3.1 Location

The concept of location serves to ask and answer *where* questions. It is the most fundamental concept of spatial information (Golledge 1995). Location is neither an attribute nor an object, but a *relation*. We locate something by relating it spatially to something else (Donnelly 2005). Thus, there is no absolute location of anything, but there are infinitely many relative ones. When talking about *a* location, one means either a place or a position (see below).

Location information relates *figures* (being located) to *grounds* (locating the figures). What gets located plays the role of *figure*, what locates it plays the role of *ground* (Talmy 1983). When we say "Santa Barbara is in California," Santa Barbara is chosen as the figure and California as the ground. Location information is always the result of a human judgment, assigning figure and ground roles to states of entities and relating them spatially. The roles of figure and ground can be assigned to states of objects (e.g., streets) or network elements (e.g., intersections).

Places are commonly used grounds. They often carry names, such as Santa Barbara, California, the Pacific Ocean, or North America. Like all grounds, they are entities in space *and* time, as the examples demonstrate. A place may start and cease to play its role as a ground—consider Yugoslavia, which was a place during a part of the 20th century. Place-based location information is normal for human communication and has become a key resource in spatial computing.

Figures get related to grounds through *spatial relations*. Locating Santa Barbara *in* California or *by* the Pacific Ocean uses prepositions to express the qualitative

relations of containment and contact; locating it by geographic coordinates uses relations expressed as angular distances from the equator and a meridian.

Positions express spatial relations quantitatively, through distances from the grounds established by coordinate systems. For example, Wikipedia[2] says that Santa Barbara has the coordinates 34° 25′ 33″N 119° 42′ 51″W, stating angular distances from the equator and prime meridian of the WGS'84 coordinate system. In the vertical dimension, heights use an ellipsoid or geoid as ground and distances from it as positions. With the grounds conventionalized through coordinate systems, positions appear to be independent entities, but they remain relations, referring to default grounds.

3.2 Field

Field information answers questions about the *value of an attribute* anywhere in a space of interest. The field concept is an elegant mathematical idea, originally developed in physics to explain gravity and widely adopted for spatial information models in geography, biomedicine, and other areas. It is, of course, the first of two fundamental views of spatial information, the second being the concept of an object (Couclelis 1992; Galton 2004). Fields capture phenomena that can be described by a property with a single value at any position in a space of interest. Geographic examples include temperature and wind fields.

Mathematically, every field is characterized by a *continuous function* from positions to values, meaning that small changes in positions map to small changes in values. The domain of the field function captures the space of interest. Positions can be spatio-temporal, although time often gets separated from space and modeled as snapshots. The positions as well as the values can be discrete, still allowing for continuous functions between them (Rosenfeld 1986).

In practice, the continuity condition on the function is sometimes ignored, retaining only the functional relationship between positions and values. Land cover or land ownership are examples of phenomena requiring this broader definition, as their values do not change smoothly. Such a generalized "field" concept has been captured in the *geo-atom* of (Goodchild et al. 2007) and standardized in the *coverage* specification of the Open Geospatial Consortium as "digital spatial information representing space-time varying phenomena" (Baumann 2010).

3.3 Object

Object information answers questions about properties and relations of objects. Objects provide the second fundamental way, after fields, of understanding spatial

[2] http://en.wikipedia.org/wiki/Santa_Barbara,_California.

information. They capture individual things extended in space that can be identified and described by properties and relations. Typical geographic examples are buildings and lakes.

The only defining characteristic of all kinds of objects is that they have an *identity*. This allows for tracking their properties and relations over time. Changes can occur in all properties and relations, including spatial relations in the case of mobile objects.

Objects are *bounded*, i.e., they have finite sizes, although their boundaries are not always known or even knowable. Many natural objects do not have crisp boundaries and are better characterized by transition zones between what clearly belongs to them and what does not (Burrough and Frank 1996). Consider mountains, forests, beaches, or ash clouds; while some positions clearly lie within these objects, others clearly lie outside, and the transition zones are often vast. Boundaries, crisp or vague, may or may not be necessary for analysis.

The *properties* of objects are attributed to them as a whole and *relations* between objects hold for them as wholes. All properties and relations are either spatial or thematic; temporal aspects of objects (such as their creation) are captured by events in which objects participate. The values of an object's properties and the relations it participates in define its *state* at any time.

Parts of objects can themselves be treated as objects, and *complex objects* get aggregated from simpler ones. *Feature*s are parts of the surfaces of objects and can be considered special cases of objects. For example, while lakes may be seen as three-dimensional objects, they are also often treated as two-dimensional parts of the earth's surface.

The large variety of objects in any domain suggests a need to classify them. Object *types* are typically defined based on shared properties and relations. For example, water bodies may be classified into those with standing or flowing water, with lakes as a further subclass of standing water bodies.

3.4 Network

Networks hold information on connections between objects. They are used to answer a broad range of questions about connectivity, such as whether an object is reachable from another object, what the shortest path is between them, how central an object is in a network, where the sources and sinks are of something flowing through the network, or how fast something will spread and where it will spread to. Network applications benefit from modeling networks as graphs and from the vast choice of algorithms and implementations coming with this. As a consequence, the concept of networks is most broadly applied across disciplines.

The objects connected through a network are called its *nodes*. The variety of objects that can play the role of nodes is unlimited. They include physical objects (such as people or places), as well as mental and social ones (such as concepts, web pages, or companies).

Nodes get connected by any binary relation of interest, forming a network's *edges*. Every pair of nodes participating in the relation is connected by an edge. Edges may have physical realizations, as do roads, or they may be modeled as abstract connections only, as in the case of social relations. Edges, and with them whole networks, can be *directed*. For example, a road segment may be seen as a directed connection from one place to another. Applications often use a single attribute to characterize a relevant property of edges. A numeric attribute is called the *weight* or *impedance* of the edge, a nominal attribute its *color*. For example, travel distance (measured by length or time) can serve as weight, with longer connections having greater weights; colors can express different types of edges, such as travel modalities (walking, driving).

A *path* in a network is a sequence of nodes where each consecutive pair of nodes is connected by an edge. A node is reachable from another node, if there is a path connecting the two. A path returning to its origin is a cycle. Sometimes it is useful to think of a path as the sequence of edges connecting the nodes. A network is connected if it has a path between every pair of nodes.

A network can be *embedded* in a surface or a three-dimensional space, so that its nodes have positions and its edges shapes. Embeddings on the earth's surface or in a building are typical geographic examples. Many more network properties are defined in the literature—see, for example, (Newman 2010).

3.5 Event

Event information answers questions about what has happened, is happening, or may happen. For example, one might want to know where two people first met, how long it takes to get through a traffic jam, or whether it will rain tomorrow. While static maps show snapshots of unfolding events, computer models and their visualizations can now represent events and relationships between them. Events are rapidly becoming an important subject of spatial information, reflecting the fact that information about changes in our dynamic world is increasingly available, even in real time.

Events are individual portions of processes. For example, considering weather and traffic as processes, a rainstorm is an individual weather occurrence and a traffic jam is a single, limited occurrence of heavy traffic. All events are temporally bounded, i.e., of finite duration. Their beginnings and endings are not always known, just like for the spatial boundedness of objects.

Events have an *identity* and are described by temporal and thematic properties and relations. For example, a rainstorm may be named and characterized by its duration and the accumulated rainfall. Commonly used *temporal properties* of events are their duration or temporal bounds. The main temporal *relations* between events are precedence, co-occurrence, and posteriority. For example, a rainstorm can precede a traffic jam, co-occur with an electricity blackout, and follow (be posterior to) gusts of wind.

The key *relation* between events and the other core concepts is that of *participation*. Events involve fields, objects, and networks as participants, which often get changed through their participation. For example, rainstorms change temperature and humidity in their area, and traffic jams affect the choice of routes through road networks.

Parts of events can themselves be treated as events, and *complex events* can be aggregated from simpler ones. Part-whole relations are as central to event models as they are to object models. For example, a particular rainstorm may be part of a storm system sweeping a region and there may be questions about rainfalls or damages in individual storms as well as about their overall accumulation.

3.6 Granularity

Granularity information answers questions about the amount of detail in spatial information. For example, one may want to know how precisely a smart phone can locate itself, what cell size underlies a global climate model, what the smallest recorded buildings are on a map, whether gravel roads are included in a road network, or how long a snow storm must last to be recorded as a blizzard. Granularity characterizes all content concepts of spatial information.

Granularity is a concept of information *quality*, used to describe information itself rather than things in the world. Every piece of spatial information comes with some granularity, implicitly or explicitly, i.e., with a level of detail it captures. While the term granularity is sometimes used in a narrower sense to describe the level of detail in conceptualizations (but not in representations external to the mind), it is used here to mean the level of detail affecting any stage of producing or using spatial information—starting with conceptualization, and going through observation, representation, integration, and analysis, to visualization.

Commonly used alternative terms for granularity, in addition to level of detail, are *resolution* and *precision*. The term *scale* has so many different and often conflicting meanings that it is best used as an informal notion only, but it refers primarily to granularity. Other related terms, for instance *discrimination* or *precision* (in its second sense of repeatability of measurement), refer to measuring instruments, not to information, but they impose lower limits on granularity. Finally, the terms *spacing* and *support* have more specific technical meanings in measurement processes.

Spatial and *temporal* granularity are extents in space and time and therefore, in principle, measurable quantities. Yet, variations throughout the observation-to-visualization life cycle of information make it hard to come up with a generally applicable measure for granularity. Given that all information is rooted in observation, granularity is best understood starting from observations and then defining measures for derived information, for example for field or object information. When talking about observations, granularity is usually referred to as *resolution* (Degbelo and Kuhn 2012). It results from the fact that the change observed through observation necessarily has a minimum below which it cannot be detected. *Thematic*

granularity is considerably harder to formalize. There is not always a clear ordering of thematic classes, nor is there always a numeric measure for thematic granularity.

3.7 Accuracy

Information is accurate if it describes something correctly. Accuracy can only be determined for a given granularity and with respect to some reference information that is considered (more) accurate. Accuracy has been a long-standing concern for the theory and practice of spatial information (Burrough and McDonnell 1998). While the emphasis has often been on locational accuracy, the concept of accuracy applies to all information about fields, objects, networks, and events. Accuracy and granularity together are the main indicators of the quality (or certainty) of spatial information.

The degree of accuracy is measured by the difference between the reference information and the information in question, either statistically or for individual values. For example, one may determine the accuracy of a building footprint obtained from aerial photography by comparing it to ground survey data. Inaccuracy in information results from errors in observation or analysis. Accuracy is limited by observation procedures and gets propagated through analysis.

The difference between the mean of repeated observations and a hypothetical true value is called *bias*. With adequate measuring equipment, the results of repeated measurements or calculations distribute regularly around the true value. This property is a consequence of understanding measurement as a random process.

Accuracy is often associated with the absence of *systematic* errors, such that measurement is affected only by *random* errors. Systematic errors get minimized or eliminated by calibration, which should reduce systematic errors to a level below the granularity of the information. As a consequence, best practice of reporting information requires stating only as many digits after the decimal as the combined effect of granularity and (in)accuracy permit.

4 Core Spatial Computations

The ambition behind defining core concepts was to reduce the complexity of spatial information to a few powerful notions that are meaningful across applications. The practical impact of this idea, however, comes from reducing the complexity of *computations* to a similarly low number. For this purpose, simply reorganizing existing GIS commands around the core concepts would not be sufficient. Instead, the approach taken in this work is to define *core computations* for each core concept. These operators constitute the semantic primitives of our language of spatial computing, which can then be combined to express more complex computations. For example, questions about the location of objects get answered by combining the operators for Location and Object. The operators can take many

syntactic forms, depending on the chosen embedding of the language. This section presents a simplified algebraic specification derived from a Haskell embedding; the next section presents a Python embedding.

The length of this paper does not allow for a detailed discussion of the operators, but they should be largely self-explanatory for GIScience researchers. For a better understanding, readers should consult the concept descriptions in the previous section. We present the operator signatures in the form of a table, omitting the Haskell axioms specifying their semantics (see Table 2). The signatures are written following a standard practice in software engineering: the name for the operator followed by parameters for the input types (combined as a cross product) and for the output type (after the arrow). Note that the type parameters are further constrained in the full specifications. For example, the figure and ground parameters can be instantiated by objects, nodes, or edges.

The choice and development of the operators is the subject of ongoing research. Specifications for the quality concepts (granularity and accuracy) are still being developed. The main goal of producing embeddings like those in Haskell and Python is in fact to test the operators for completeness, consistency, and adequacy in practical GIS projects.

5 A Python Embedding

To demonstrate applicability in the current technological context, we outline a Python implementation of the proposed language for spatial computing. Python, created in the 1990s in response to the verbosity and complexity of object-oriented languages such as C++, is a mature general-purpose, multi-paradigm programming language used in industry and academia in domains ranging from Web development to high-performance scientific computing. Libraries such as SciPy and PySAL[3] include spatial computing tools, and well-known spatial libraries such as GDAL provide Python bindings.[4] Current GIS display a high degree of interoperability with Python, providing APIs and bindings at multiple levels. Notably, ArcGIS[5] and QGIS[6] have adopted Python as their principal scripting language, making it a suitable testing ground for the proposed approach to core concepts. More generally, the popularity of Python among researchers and practitioners in many domains lowers the adoption barrier of the proposed approach. The source code is available online at http://github.com/spatial-ucsb/ConceptsOfSpatialInformation.

The first building block for our domain-specific language consists of its abstract data types (ADT). Starting from the Haskell specifications of core concepts, we

[3] https://pysal.readthedocs.org, http://www.scipy.org.

[4] http://gdal.org/python.

[5] http://resources.arcgis.com/en/communities/python.

[6] http://pyqgis.org.

Table 2 Overview of core computations on spatial information

	Operators	Comments
Location	isAt: figure × ground → Bool	The contact relation
	isIn: figure × ground → Bool	The containment relation
	position: figure → point	A point positioning the figure
	bounds: figure → shape	A shape bounding the figure
Field	new: [(pos,val)] × ([(pos × val)] × pos → val) → field	Interpolating a field from a list of values
	bounds: field → shape	The domain of the field
	getValue: field × pos → val	The value at a position
	local: field × (val → val') → field	Computing new values at all positions
	focal: field × (pos → val') → field	Computing new values from neighborhoods
	zonal: field × (pos → val') → field	Computing new values from zones
Object	get: object × (object → value) → value	Get the value of a property of the object
	is: object × object × (object × object → Bool) → Bool	Are the two objects in the given relation?
	same: object × object → Bool	Are the two objects the same?
Network	nodes: network → [node]	All nodes in a network
	edges: network → [edge]	All edges in a network
	addNode: network × node → network	Add a node to a network
	addEdge: network × edge → network	Add an edge to a network
	degree: network → node → Int	Degree of a node in the network
	connected: network × node × node → Bool	Are two nodes connected?
	shortestpath: network × node × node → [node]	The shortest path between two nodes
	distance: network × node × node → Int	The network distance (as number of nodes)
	breadthfirst:: network × node × Int → [node]	All nodes at distance from a node
Event	when: event → date	The time of an event as a date
	within: event → period	The time of an event as a period
	same: event × event → Bool	Are the two events the same?
	during: event × event → Bool	Is the first event happening during the second?
	before: event × event → Bool	Is the first event happening before the second?
	after: event × event → Bool	Is the first event happening after the second?
	overlap: event × event → Bool	Does the first event overlap the second?

defined a set of Python classes, bearing in mind the semantic differences between these two languages. Python is strongly dynamically typed, favors procedural and object-oriented programming, and does not provide a mechanism for type parameterization. By contrast, Haskell is a functional language and strongly statically typed, i.e. all types are known and checked at compile-time. In Python, ADTs are not definable directly, and have to be embedded in concrete *classes*, while Haskell *type classes* are abstract interfaces, providing powerful support for type parameterization.

The proposed Python classes are a lower level of abstraction, between the Haskell specifications and the concrete software components that perform the spatial computation. This way, existing software resources can be harnessed and deployed in the computational workflow, without locking the user into a software-driven conceptualization of their domain knowledge. To provide an illustration of the approach, the Python class that defines the *field* concept is structured as follows (the "Cc" prefix stands for "core concepts"):

```
class CcField(object):

    def getValue( self, position ):
        """
        @param position a position in the field
        @return the value of field at position
        """
        raise NotImplementedError("getValue")

    def local( self, fun ):
        """
        Map algebra's local operations, with a
               function to compute the new values
        @param fun a function to be locally applied on the field
        @return a new CcField field
        """
        raise NotImplementedError("local")

    def focal( self, fun ):
        """
        Map algebra's focal operations, with a kernel function to
               compute the new values based on the neighborhood of the position
        @param fun a Kernel function to be applied on the field
        @return new CcField field
        """
        raise NotImplementedError("focal")

    def zonal( self, fun ):
        """
        Map algebra's zonal operations, with a
               function to compute the new values
               based on zones containing the positions.
        @param fun a function to be zonally applied on the field
        @return new CcField field
        """
        raise NotImplementedError("zonal")
    ...
```

This class includes a getter (*getValue*), and standard Map Algebra operations. As Python does not provide an explicit mechanism to define abstract classes, we

simulate the abstract nature of the class by raising exceptions at run-time (i.e. *NotImplementedError*), highlighting the fact that the class should not be instantiated directly. This class can be implemented to interface with the technological layer, in this case by handling the popular GeoTiff raster format,[7] reusing the logic defined in the software package GDAL[8]:

```
import gdal # import implementation library

class GeoTiffGdalField(CcField): # subclass of CcField

    def __init__(self, file_path):
        # load GeoTiff from file using GDAL
        self.gdalHandler = gdal.Open(file_path, GA_Update)
    def getValue(self, position):
        # use GDAL to retrieve field value for position
        return self.gdalHandler.ReadAsArray( position.x, position.y, ... )

    ...
```

Similarly, ArcMap libraries can be harnessed through Python in an appropriate subclass:

```
import arcpy # import implementation libraries
from arcpy import sa

class GeoTiffArcMapField(CcField): # subclass of CcField
    def __init__(self, file_path):
        # load GeoTiff from file using ArcMap
        self.arcRaster = sa.Raster(file_path)
    def getValue(self, position):
        # use ArcMap to retrieve field value for position
        result = arcpy.GetCellValue_management( ... position.x, position.y )
        return result.getValue(0)
    ...
```

These concrete classes act as wrappers, encapsulating the implementation details. This approach enables the user to define operations and queries on fields, only relying on the *CcField* interface, hiding the technological and data layer from view. Similarly, we define the concepts *object* and *network* as:

[7] http://trac.osgeo.org/geotiff.

[8] Created by the Open Source Geospatial Foundation, the Geospatial Data Abstraction Library (GDAL, http://www.gdal.org) is a software package that provides an interoperability layer between a variety of raster and vector data formats.

```
class CcObject(object):
    def bounds( self ):
        """ @return geometric bounds of the object """
        raise NotImplementedError("bounds")

    def relation( self, obj, relType ):
        """ @return Boolean True if self and object obj are in
                    a relationship of type relType """
        raise NotImplementedError("relation")

    def property( self, prop ):
        """ @param prop the property name
            @return value of property in object """
        raise NotImplementedError("property")

    def identity( self, obj ):
        """ @param obj an object
            @return Boolean True if self and obj are identical """
        raise NotImplementedError("identity")

class CcNetwork(object):
    def nodes( self ):
        """ @return a copy of the graph nodes in a list """
        raise NotImplementedError("nodes")

    def edges( self ):
        """ @return list of edges """
        raise NotImplementedError("edges")

    def addNode( self, n ):
        """ Add a single node n """
        raise NotImplementedError("addNode")

    def addEdge( self, u, v ):
        """ Add an edge between u and v """
        raise NotImplementedError("addEdge")

    ...
```

Existing efficient vector and network data manipulation libraries, such as GDAL and NetworkX,[9] can be tapped in the corresponding implementations. Once the set of core concepts have been implemented and linked to the technological layer, the user can perform spatial computations directly in terms of the concepts.

Selecting the analysis of solar energy collection potentials as a test domain of spatial information, a typical resource consists of Shapefiles with detailed vector data representing building roofs and other viable areas for the installation of solar panels. Using the Python core concepts, the user can load these objects and perform spatial operations on them. In the following example, the user formulates the question: *is the roof of the Poultry building located in a viable area?*

```
import shapefileToObjects from coreconcepts

roofs = shapefileToObjects( "data/Rooftops.shp" ) # load objects
viableAreas = shapefileToObjects( "data/Vareas.shp" ) # load objects
polRoof = roofs[2] # get the roof for Poultry Science building from the array
print polRoof.property('name') # prints "Poultry Science"
# find answer to question
valid = any(map(lambda area: polRoof.relation(area,'within'), viableAreas))
```

[9] https://networkx.github.io.

Because of their foundational nature, the core concepts can be deployed and assembled to represent a wide variety of domain entities, enabling the user to flexibly model their scenarios and even include multiple conceptualizations. Thus, the energy analyst can define a roof type as a field for some purposes (e.g., modeling its topography) *and* as an object for others, formulating spatial questions that involve both perspectives:

```
class Roof(CcField,CcObject):
    ...

roofA = Roof( 'some_data_source' ) # load a roof
roofB = Roof( 'some_data_source' ) # load another roof

# does roofA have a smaller area than roofB?
answer = roofA.property('area') < roofB.property('area')
# is value of field position (3,5) in roofA higher than
# half of the same position in roofB?
answer = roofA.getValue([3,5]) > roofB.getValue([3,5])/2
```

As we have shown in this section, the embedding of core concepts in Python can provide a widely usable and modular conceptual layer to organize domain-specific spatial knowledge. The advantages of this approach will be particularly evident in the context of information integration from different domains, providing a modeling framework as well as a computing toolkit to facilitate communication between GIS practitioners and domain experts.

6 Conclusions and Outlook

We have outlined the design rationale and an early implementation of a language for the "domain" of spatial computing. Our ultimate goal is a high-level language executable on existing commercial and open source spatial computing platforms, in particular Geographic Information Systems (GIS). So far, we have specified a set of core concepts and, for each of them, a set of core computations. These specifications are now being translated (by hand) into Python scripts, as well as into Haskell data types and foreign function calls, in both cases allowing for calls to commercial or open source GIS platforms.

The paper first described the theory of core concepts of spatial information that underpins the language, which includes five core content concepts (*location*, *field*, *object*, *network*, and *event*), complemented by two core quality concepts (*granularity* and *accuracy*). After providing a formal specification of a set of spatial computations relying on these concepts, we outlined an ongoing embedding in Python, showing how this package can function as a mediation layer between the

core concepts and existing technological layers that encode the data and perform the computations.

More interdisciplinary research is needed to achieve our vision. The Python embedding will be stress-tested in more realistic scenarios, providing feedback to revise both the formal specifications and the software embedding, but possibly also the concept selection. Use cases set in different domains, ranging from ecology to economics and history, will help demonstrate the cross-domain transferability of the core concepts within and beyond the traditional scope of GIS applications.

A different route to take with this idea is to design and implement Application Programming Interfaces (API) on top of spatial data repositories and spatial database management systems. For example, popular open data like OpenStreetMap or the US Census TIGER data would benefit from some generic computing layer through which to query and analyze them from certain perspectives (for example, seeing them as representing networks or sets of objects).

Acknowledgments We gratefully acknowledge contributions to the Python embedding and testing from Michel Zimmer, Marc Tim Thiemann, and Eric Ahlgren as well as funding from the UCSB Center for Spatial Studies.

References

Albrecht, J. (1998). Universal analytical GIS operations: A task-oriented systematization of data structure-independent GIS functionality. In H. Onsrud & M. Craglia (Eds.), *Geographic information research: Transatlantic perspectives* (pp. 577–591). London: Taylor & Francis.

Baumann, P. (2010). The OGC web coverage processing service (WCPS) standard. *Geoinformatica, 14*(4), 447–479.

Burrough, P. A., & Frank, A. U. (1996). *Geographic objects with indeterminate boundaries*. London: Taylor & Francis.

Burrough, P. A., & McDonnell, R. (1998). *Principles of geographical information systems*. Oxford, UK: Oxford University Press.

Camara, G., Egenhofer, M. J., Ferreira, K., Andrade, P., Queiroz, G., Sanchez, A., et al. (2014). Fields as a generic data type for big spatial data. In *Geographic Information Science* (pp. 159–172). Berlin: Springer.

Couclelis, H. (1992). People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. In A. U. Frank, I. Campari, & U. Formentini (Eds.), *Theories and methods of spatio-temporal reasoning in geographic space* (pp. 65–77). Berlin: Springer.

Degbelo, A., & Kuhn, W. (2012). A Conceptual Analysis of Resolution. In *GeoInfo—XIII Brazilian Symposium on GeoInformatics, November 25–28 2012, Campos do Jordão, Brasil* (pp. 11–22).

Donnelly, M. (2005). Relative Places. *Applied Ontology, 1*, 55–75.

Egenhofer, M. J., & Kuhn, W. (1999). Interacting with Geographic Information Systems. In M. F. Goodchild, D. J. Maguire, D. W. Rhind, & P. Longley (Eds.), *Geographical Information Systems: Principles, techniques, applications, and management* (2nd ed., Vol. 1, pp. 401–412). New York: Wiley.

Galton, A. (2004). Fields and objects in space, time, and space-time. *Spatial Cognition & Computation, 4*(1), 39–68.

Ghosh, D. (2011). DSL for the uninitiated. *Communications of the ACM, 54*(7), 44.

Golledge, R. G. (1995). Primitives of spatial knowledge. In T. L. Nyerges, D. M. Mark, R. Laurini, & M. J. Egenhofer (Eds.), *Cognitive aspects of human-computer interaction for geographic information systems* (pp. 29–44). Berlin: Springer.

Goodchild, M. F., Yuan, M., & Cova, T. J. (2007). Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science, 21*(3), 239–260.

Janelle, D. G., & Goodchild, M. F. (2011). *Concepts, principles, tools, and challenges in spatially integrated social science*. SAGE Publications: In the SAGE Handbook of GIS and Society.

Kuhn, W. (2012). Core concepts of spatial information for transdisciplinary research. *International Journal of Geographical Information Science*, *26*(12), 2267–2276 (Special Issue in honor of Michael Goodchild).

Newman, M. E. J. (2010). *Networks*. Oxford: Oxford University Press.

Norman, D. A. (1986). Cognitive Engineering. In D. Norman & S. Draper (Eds.), *User centered system design* (pp. 31–61). Hillsdale, NJ: Lawrence Erlbaum Associates.

Rosenfeld, A. (1986). "Continuous" functions on digital pictures. *Pattern Recognition Letters, 4*(3), 177–184.

Talmy, L. (1983). How language structures space. In H. L. Pick & L. P. Acredolo (Eds.), *Spatial Orientation* (pp. 225–282). New York/London: Plenum Press.

Drawing with Geography

Takeshi Shirabe

Abstract A method is proposed to assist spatial planners in drawing with 'geographic' constraints. These constraints constrain graphic objects to have certain relationships that are not limited to be (Euclidean) geometric or topological but allowed to be dependent on the spatial variation of selected conditions (e.g., elevation and vegetation) characterizing an underlying geographic space. Just as in existing computer-aided design systems, the method accepts a manual change to a graphic object or constraint, and updates all affected graphic objects accordingly. The paper discusses how such a method is motivated and improves the graphic editing capability of geographic information systems, and identifies key issues for its implementation.

Keywords Constraint-based drawing · Geographic information systems · Computer-aided design · Geographic constraints · Geodesign

1 Motivation

This work has been motivated by seemingly naïve questions that have arisen from informal departmental discussions on architectural design and spatial planning: Both architects and spatial planners design space (though of different scopes and scales), and the use of computer-aided design (CAD) has become commonplace in architectural design practice, but not in spatial planning practice. Why not? What about geographic information systems (GIS), which store, process, and present spatially-referenced data? Can't they be spatial planners' CAD?

Spatial planning is a process of decision making on how limited land resources should be used to satisfy present and future demands within a given jurisdiction at

T. Shirabe (✉)
School of Architecture and the Built Environment, KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: shirabe@kth.se

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_19

reasonable economic, social, and environmental costs. A planning project involves many non-graphically-oriented tasks ranging from analyses of demographic changes and economic trends to public hearings and negotiations with stakeholders. An outcome, however, often takes a (carto)graphic form in which geographic features such as corridors and regions are drawn and designated for certain land uses such as highway alignment and nature conservation. Thus, if geographic features are effectively abstracted to graphic objects, GIS indeed help spatial planners manipulate them by design.

Existing GIS and CAD allow users to edit graphic objects in a given digital geographic space by specifying coordinates of which they consist and/or constraints to which they are subject. These systems support at least three types of geometry: point, line, and area. A point is a zero-dimensional object with its location identified by a pair of x, y coordinates. A line is a one-dimensional object consisting of a sequence of line segments, each connecting two points. An area is a two-dimensional object bounded by a closed sequence of line segments. Because of this piecewise linear approximation, line and area are sometimes referred to as polyline and polygon, respectively (although some GIS are capable to store graphic objects having a regular shape, e.g., arc, spline curve, circle, or oval, exactly using mathematical formulas). Figure 1 illustrates on the left a sample drawing of a line connecting two disks through points.

Typical constraints concern sizes (e.g., length and area) of graphic objects or relationships (e.g., distance and parallelism) between graphic objects. For instance, GIS and CAD may be used to create a line segment by selecting its origin, specifying its length, and associating with it a constraint such that the line segment is parallel to another line segment. Furthermore, these systems can modify graphic objects automatically, in response to a modification of a graphic object to which

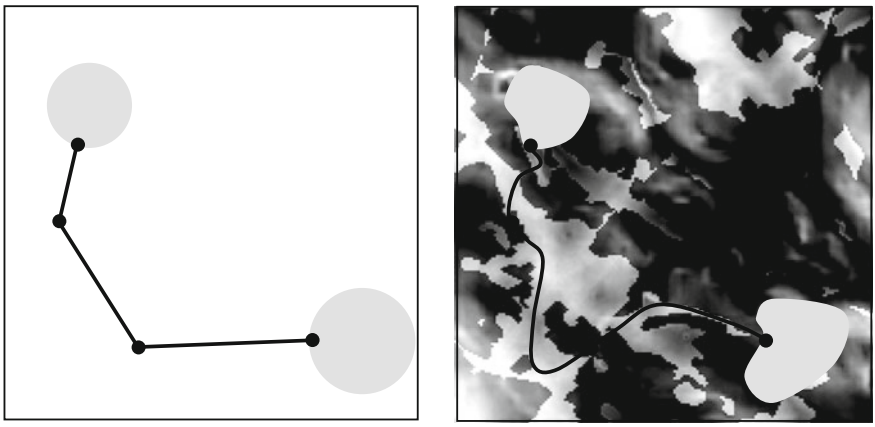


Fig. 1 Sample drawing of a line (*solid line*) connecting two disks (*gray areas*) through points (*black dots*), situated on Euclidean plane (*left*) and on in a geographic space (*right*) where the cost to place each graphic object varies from location to location (the *darker shading* represents the higher cost)

they are constrained. For example, if one of the aforementioned parallel line segments is rotated, then the other will be rotated automatically so that the two line segments remain parallel.

The process of graphic editing subject to geometric constraints is commonly known as "constraint-based drawing" (Gleicher and Witkin 1994). A variety of methods have been developed for transforming constraint data to coordinate data (see, e.g., Bettig and Hoffmann (2011) for a survey). For these methods to work, it is typically assumed that graphic objects are embedded in Euclidean space and their sizes or relationships are defined in terms of Euclidean geometry.

In professions that involve analysis and synthesis of geographic space, however, some graphic objects may well possess sizes and relationships that are not Euclidean geometric or topological but 'geographic'—which is here meant to be dependent on the spatial variation of selected conditions on or near the surface of the earth. For example, an environmental planner may consider the length of a corridor as the accumulation of environmental impact along that corridor. As another example, a landscape architect may relate a landfill to a residential area through visibility over a terrain.

The geographic interpretation of otherwise purely geometric properties, in turn, redefines the characters of graphic objects. To see this, suppose that the line and disks drawn on the left of Fig. 1 are now situated in a geographic space where the cost for them to be varies from location to location. Then, as illustrated on the right of Fig. 1, the line segments may no longer look straight and the disks may no longer look round.

Many raster-based GIS allow a user to specify a variety of geographic sizes and relationships for graphic objects and calculate their values in response to relocating, reshaping, or resizing the graphic objects. These systems, however, are not designed to do the inverse, that is, automatically relocate, reshape, or resize graphic objects so that they attain desired values for the specified geographic sizes and relationships. If the user still wants to do this task, he/she must (1) edit graphic objects manually, (2) calculate the values of their sizes and relationships, (3) see if those values are as desired and, if not, (4) return to the initial step of manual editing. Unfortunately, a trial-and-error approach as such could easily take a prohibitively long time to have all graphic objects satisfy all constraints.

Alternatively, one may be tempted to formulate a task of drawing graphic objects with geographic constraints as a mathematical programming problem. It concerns the aggregation of elementary spatial units (e.g., grid cells) into larger groups, each of which forms a specified type of geometric object, maximizing or minimizing specified functions subject to specified constraints. Typical applications include site selection (e.g., Cova and Church 2000; Xiao 2006), land-use planning (e.g., Aerts et al. 2003; Ligmann-Zielinska et al. 2008), and conservation reserve design (e.g., Church et al. 2003; Nalle et al. 2003; Önal and Wang 2008). Exact solution methods exist for some instances (e.g., Williams 2002; Shirabe 2005), but it has been found that they cannot deal with problems involving more than several hundreds of spatial units (see, e.g., Xiao (2006) for computational experiments). More problematic in the optimization approach is that a task of designing spatial

plans may well involve intangible factors (e.g., aesthetic quality) that do not lend themselves to mathematical formulation.

In contrast to these fully-automated approaches to spatial allocation, other approaches have cast computation in a supporting role while retaining an emphasis on more traditional forms of human decision making. The idea of "spatial decision support" was introduced to the fields of urban and regional planning (e.g., Batty and Densham 1996) and geographic information science (e.g., Jankowski 1995) decades ago and has been often embodied by linking GIS and optimization algorithms/heuristics. A typical mechanism is such that humans specify constraints or criteria, computers generate alternative solutions, and humans evaluate results and either select a solution or re-specify criteria. A major problem with this approach is that it minimizes the extent of human participation in generating solutions. While that may not diminish the quality of any particular solution (since computers will usually be better able to formally optimize), it tends to diminish the effectiveness with criteria are re-specified on subsequent iterations. Without an ability to refine those criteria with the insights gained from exploring alternative solutions, the whole allocation exercise is seldom able to achieve any more than an adequate solution—one that lacks the serendipity, the synergy, and the often-irrational human creativity.

In response to this limitation, a new paradigm of use of GIS as an interactive, integrated platform for design, analysis, and presentation of spatial plans has recently emerged among both professionals (e.g., GeoDesign Summit 2010–2014) and researchers (e.g., Spatial Concepts in GIS and Design 2008; Digital Landscape Architecture 2009–2014). The paradigm is called "GeoDesign" and, if its promise is realized, GIS users should be able to play a much more active role in creating (as opposed to evaluating) design alternatives. GeoDesign, however, is still a vision and has yet to be found how exactly GIS can or should support the creative process. To, at least partially, address this gap, this paper proposes that constraint-based drawing will be a key component of GeoDesign and discusses how it can be adapted to work with geographic constraints.

The remainder of the paper is organized as follows. Section 2 presents a method of drawing with geographic constraints in a digital geographic space. Section 3 shows an implementation of the method. Section 4 concludes the paper.

2 Method

A method of constraint-based drawing in a digital geographic space is here generally described in terms of the following aspects:

- how spatially-varying conditions are represented
- how graphic objects are represented
- how graphic objects are constrained
- how graphic objects and constraints are edited manually, and
- how graphic objects are updated automatically.

2.1 Representation

Spatially-varying Conditions
A given geographic space is discretized into a two-dimensional array of cells, and a graph structure is implicitly imposed on these cells by connecting each pair of laterally or diagonally adjacent cells with an edge. A spatially-varying condition (e.g., topographic elevation or environmental impact) over the geographic space is then represented by a "map"—in the sense of Tomlin (1990)—on which a single value is assigned to each cell. If necessary, the value of each edge, too, is derived from a map in the following manner. The value of an edge connecting two laterally adjacent cells is the arithmetic mean of the values of the two cells, and the value of an edge connecting two diagonally adjacent cells is the arithmetic mean of the values of the two cells multiplied by the square root of two.

Graphic Objects
A point is represented by a single cell and assumed to have no size. It has a state that can be changed by the user between fixed and variable. Fixed points can be (re) located only manually by the user, and variable points automatically in relation to other graphic objects (to be discussed later). As described below, points may be stand-alone or components—referred to here as 'children'—of other graphic objects.

A line segment is represented by a cellular path, i.e., a sequence of cells/edges connecting two points designated as vertices. Given a map that determines edge values, the size or length of a line segment is defined as the sum of the values of all edges composing the line segment. Two vertices are not sufficient to determine a line segment, because more than one path can connect the two vertices. Of them, the present method selects a shortest path as a geographic analogy to a Euclidean line segment. A tie is broken arbitrarily.

A line is represented by a sequence of line segments. Given a map that determines edge values, the length of a line is defined as the sum of the lengths of all line segments composing the line. To specify a line, a map and a sequence of vertices are required, from which a sequence of line segments is determined such that each line segment is a shortest path between two adjacent vertices. Note that a line and its constituting line segments always share the same map.

An area is represented by a cellular region, i.e., a set of connected cells. Given a map that determines cell values, the size of an area is defined as the sum of the values of all cells composing the area. Depending on the application domain, more specific sub-types of areas may be introduced. For example, a 'circular' area is an area such that all its cells are within a certain radius (whose length is measured using a map associated with the area) from a point designated as its center. Another example is a 'polygonal' area as a closed sequence of line segments enclosing it. Unlike the case of Euclidean geometry, however, those line segments might unexpectedly intersect each other and result in fragmented subareas. Furthermore, the scale of spatial planning is typically so large (much larger than that of

architectural design) that outlines of areas would not normally be worked out in detail, but rather arise as a consequence of other factors including their locational preferences and sizes. This calls for yet another sub-type of area which concerns function rather than form. One such example is a 'size-maximizing (or -minimizing)' area, which consists of a specified number of cells including a point designated as the area's root and maximizes (or minimizes, respectively) its size. The size-maximizing or size-minimizing sub-type responds to the frequent application of "suitability analysis" (see McHarg 1969) in spatial planning practice. This is the task to create a map representing the benefit or cost associated with a certain land use and to search it for the most suitable or least unsuitable area for that use.

Figure 2 illustrates (on the left) examples of these graphic objects, including five fixed points (P1, P2, P4, P5, and P6), two variable points (P3 and P7), two lines (L1 and L2) comprising two line segments (L23 and L34) and three line segments (L56, L67, and L74), respectively, and a circular area (labeled A1). For ease of illustration, the shape of each graphic object has been simplified.

Constraints

Graphic objects may be constrained to have certain relationships (to others) that need not be Euclidean geometric or topological but can be dependent on spatial variation of selected conditions on the underlying geography. Examples of geographic relationships include "distance," "visibility," and "flow direction."

A constraint is specified by two graphic objects (where the first one is constrained to the second one), a relationship, a map (if the relationship is evaluated with a spatially-varying condition), a sign of equality or inequality, and a bound value. These parameters together indicate that the specified relationship is bounded from above and/or below (depending on the sign) the specified value. Note that if a graphic object is subject to a constraint, all its components (e.g., sink, vertices, and

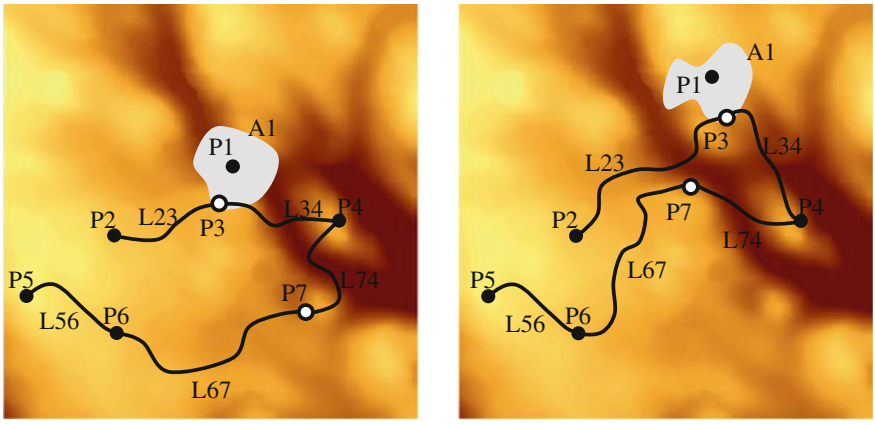


Fig. 2 Sample drawing in a geographic space characterized by maps including a map of elevation on which lightly shaded cells are higher topographic heights

line segments of a line, and center, root, and leaf of an area) are subject to the same constraint.

To illustrate how constraints are specified, suppose that the graphic objects in Fig. 2 are subject to the following constraints:

- Constraint 1: Point 3 must be inside Area 1
- Constraint 2: Point 7 must be in sight of Point 3
- Constraint 3: Point 7 must be within 30 min of walk from Line 1 (comprising Line segments 23 and 34).

Then, Constraint 1 is specified by Point 3, Area 1, the containment relationship, the equality sign, and the value of TRUE. Constraint 2 is specified by Point 7, Point 3, the visibility relationship, an elevation map from which a terrain is implied, the equality sign, and the value of TRUE. Constraint 3 is specified by Point 7, Line 1, the distance relationship, a map of walking travel time in minutes per cell (not shown in Fig. 2), the less-than-or-equal-to sign, and the value of 30.

For any relationship listed above as an example, an algorithm is available to identify which cells may be part of a graphic object to attain that relationship (see, e.g., Haverkort et al. (2009) for visibility and Jenson and Domingue (1988) for flow direction). Thus, if a graphic object is subject to one or more constraints, it is possible to extract the set of all cells from which cells may be selected to compose the graphic object without violating any of the constraints. This set is here referred to as the "drawable space" of the graphic object. For instance, Point 3's drawable space is the set of all cells composing Area 1, and Point 7's drawable space is the intersection of the set of all cells in sight of Point 3 and the set of all cells within 30 min of walk from Line 1.

In addition to user-specified constraints, all graphic objects except fixed points are implicitly subject to internally-generated constraints that follow.

- An area must contain its center
- A line segment must be incident on its vertices
- A line must consist of its line segments
- A variable point designated as a vertex of a line must be as close as possible to each of its adjacent vertices along the line; that is, the length of the corresponding line segment must be minimized. If it has more than one adjacent vertex (which may be along more than one line), the sum of the length of each corresponding line segment must be minimized.

The first three constraints are implied by the definitions of the corresponding graphic objects. The last constraint is introduced to keep the line shortest, so that the user can incrementally extend it by adding one vertex after another until a desired length is achieved.

Examples of these internally-generated constraints are given below with reference to Fig. 2.

- Constraint 4: Area 1 must contain Point 1
- Constraint 5: Point 3 must be as close as possible to Point 2 along Line segment 23 and to Point 4 along Line segment 34
- Constraint 6: Line segment 23 must be incident on Points 2 and 3
- Constraint 7: Line segment 34 must be incident on Points 3 and 4
- Constraint 8: Line 1 must comprise Line segments 23 and 34
- Constraint 9: Point 7 must be as close as possible to Point 6 along Line segment 67 and to Point 4 along Line segment 74
- Constraint 10: Line segment 56 is incident on Points 5 and 6
- Constraint 11: Line segment 67 is incident on Points 6 and 7
- Constraint 12: Line segment 74 is incident on Points 7 and 4
- Constraint 13: Line 2 comprises Line segments 56, 67, and 74.

Whether user-specified or internally-generated, all constraints are maintained using a directed "constraint (propagation) graph" (see, e.g., Kondo 1990; Owen 1991) in which a node represents a graphic object and an arc represents a constraint where its head graphic object is constrained to its tail graphic object. As illustrated in Fig. 3, a constraint graph is useful to identify which graphic objects are affected by a change to a graphic object or a constraint. For example, relocation of Point 1 will affect Area 1, Point 3, Line segments 23 and 34, Line 1, Point 7, Line segments 67 and 74, and Line 2. As a result, the drawing will be updated to the one on the right of Fig. 2.

2.2 Manual Editing

Graphic objects must be created before being edited. A new point, line, or area is created in response to explicit user request. A new line segment, on the other hand, is created when a point is added to a line as a vertex.

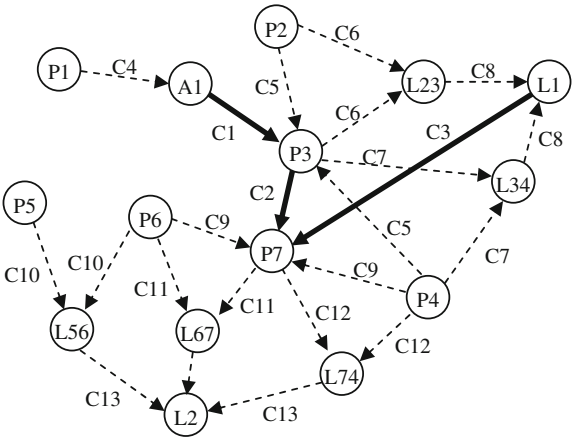


Fig. 3 Constraint graph for the graphic objects in Fig. 2. The nodes represent graphic objects. The *solid arcs* and the *dashed arcs* represent user-specified constraints (1–3) and internally-generated constraints (4–13), respectively

Constraints, too, must be created before being edited. A user-specified constraint is created in response to user request including selection of a graphic object constrained by the constraint. Internally-generated constraints, on the other hand, are created when a point is added to a line as a vertex or designated as the center or root of an area.

The method accepts user input to change parameters of existing graphic objects or constraints. If this involves (re)location or selection of a point, a pointing device such as a mouse or stylus may be used to enhance interactivity and intuitiveness. For manipulation of other, non-spatial parameters, it may be convenient to associate with each graphic object a window with graphical user interface (GUI) controls. For example, a radio button can be used for selection of a state of a point and a dropdown list for selection of a map of a line, area, or constraint.

2.3 Automatic Update

In response to a manual change to a graphic object or a constraint, all affected graphic objects are automatically updated. This may be done through an iterative procedure comprising the following steps.

The first step selects a graphic object whose parameters were changed or a graphic object constrained by a constraint whose parameters were changed. If the graphic object is a line or area, its children must be selected, too.

In Step 2, if no graphic object is selected, the procedure terminates. Otherwise, after its drawable space is updated, each selected graphic object is updated by one of the following operations depending on its geometric type.

- Operation 1 updates a point. If its state is fixed, keep its location unchanged. If its state is variable and it is designated as a vertex of one or more lines, set its location to a cell that minimizes the sum of the lengths of all line segments incident on it. Otherwise, set its location to a cell arbitrarily selected from its drawable space.
- Operation 2 updates a line segment by setting its path to a shortest path between its vertices.
- Operation 3 updates a line by arranging its line segments in the same order as its vertices.
- Operation 4 updates an area by selecting, from its drawable space, a connected set of cells. Note that the current implementation (see Sect. 3) supports only a circular area type, which consists of all cells spanned by a shortest path tree from its center.

In Step 3, select all graphic objects that are, explicitly or implicitly, constrained to at least one graphic object updated in Step 2, and repeat Step 2.

The procedure is guaranteed to terminate in a finite number of steps, since no cyclic constraints are allowed. If at least one graphic object is found not able to exist in its feasible space, however, the procedure will terminate prematurely and the

initial manual change and all the subsequent automatic updates will be undone. This happens when a graphic object of any type has an empty drawable space, or when a line or area has a non-empty drawable space such that no connected subset of it intersects the drawable space of every child of the line or area, which means that the line or area could not be connected.

To make it computationally more efficient in practice, the procedure can be modified such that all affected graphic objects are first selected (without being updated) and stored in a queue, and then updated in a "topological order" (Ahuja et al. 1993). In this way, no graphic object (and its feasible space) will be updated more than once.

3 Implementation

To show the spirit of the method without causing excessive computation, a prototype system with selected aspects has been implemented. It allows the user to create three types of graphic objects, point, line, and circular area, and to constrain points to other graphic objects through two types of relationships, containment and distance. The user can interact with the system, through a graphical user interface shown in Fig. 4, as described below.

To create a new graphic object, the user selects from the main menu an item indicating a graphic object type—"Point," "Curve," or "Region" corresponding to point, line, or circular area. The created graphic object is added to a list located to the left. To load a map, the user selects from the main menu an item labeled with "Map." Once it is loaded, the map is added to a list located at the bottom. The drawing area at the center always displays the map that is currently selected in the map list.

By selecting a graphic object from the graphic-object list, the user can see its parameters in a window located at the bottom-left corner, and, if necessary, modify them in this window. For example, the user currently selects a map called "cost.txt" over the two other loaded maps and associates it with a circular area called "Region 1" for the computation of its radius.

At any moment in a drawing session, the user may select any graphic object from the graphic-object list to be editable. Then, he/she edits it by adding or (re) locating a point with a mouse (or a stylus pen or a finger). Specific examples of this include:

- The user can set the location of an editable fixed point by clicking on a cell.
- The use can relocate an editable fixed point by dragging and dropping it to a cell.
- While a line is editable, if the user clicks on a cell, a new point will be created at the cell and added to the line as a vertex.
- While an area with its center set to null is editable, if the user clicks on a cell, a new point will be created at the cell and the area's center is set to the point.

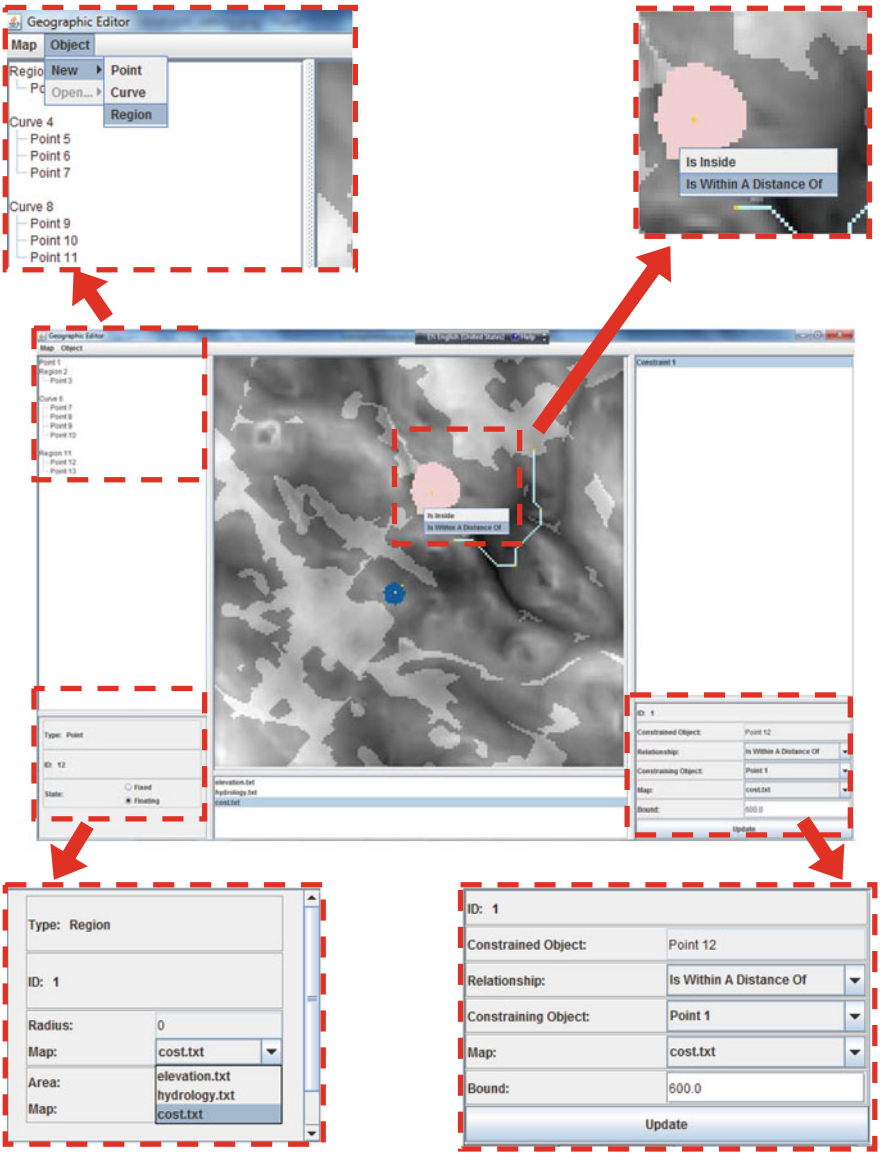


Fig. 4 Graphical user interface of a prototype system

- While an area with its center not set to null but its leaf set to null is editable, if the user clicks on a cell, a point will be created at the cell and the area's leaf is set to the point. Then the area's radius will be set to the shortest path distance between the area's center and leaf, and remain valid even if the leaf is deleted.

A constraint, too, is created interactively in the drawing area. When a point (possibly as a child of a line or area) is editable, a right-click on another graphic object will bring a context menu from which the user will select an item (either "is inside" or "is within a distance of") indicating how the point is constrained to the graphic object, and creates a new constraint accordingly. The system makes the constraint immediately effective by setting its map to the one that is currently selected in the map list and its bound to a default value.

All constraints constraining the graphic object that is currently selected in the graphic-object list are shown in a list located to the right. By selecting one from the constraint list, the user can see its parameters in a window located at the bottom-right corner, and, if necessary, modify them in this window.

Most significantly, the system always keeps every graphic object in its feasible space, every line segment connected by a shortest path, and every circular area spanned by a shortest path tree from its center without exceeding its specified radius.

While the system is still in the early stage of development, it shows how the user can develop a spatial plan by creating graphic objects and constraining them to each other in a digital geographic space. One notable aspect is that the user can explore many different scenarios informally and rapidly in the analogy of working on a traditional drafting board.

4 Discussion

This section discusses several issues learned from the current implementation of the method. First, the complexity of the area-update operation (Operation 4) varies depending on the area's sub-type. Circular areas (and polygonal areas) can be created simply by using a classic shortest path algorithm. On the other hand, the creation of area-maximizing areas needs to solve an NP-complete problem. This is the reason it is not included in the present implementation. The problem is equivalent to the "maximum value region problem" (Woeginger 1992), which is to select, from a raster map, a connected set of a specified number of cells that maximizes the sum of the values of the selected cells. If exact solutions can be compromised, however, a polynomial-time heuristic (Shirabe 2011) may be employed.

Second and related to the first issue, while a simple summation function is currently used for computation of the sizes of lines or areas, the method in theory does not preclude the use of other functions. For example, a decent bike path may be represented by a line that minimizes the highest degree of slope along the line, and a good conservation park may be an area that maximizes the lowest suitability value within that area. Interestingly, the solution of these mini-max or maxi-min problems is trivial (Shirabe 2009). Another, more geographic alternative is a "minimum work" function for a line (Shirabe 2008). It calculates the minimum amount of work it takes a wheeled vehicle to move along the line with explicit

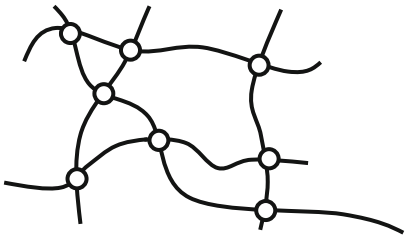


Fig. 5 Adjacent variable points (*white dots*)

consideration of the effects of gravity and friction using two maps representing the level of gravitational potential energy and the amount of energy loss due to friction. As such, more specific contexts may require more specific functions and more than one map, for computing sizes of graphic objects. And, this is true, too, for evaluating relationships between graphic objects. This indicates greater applicability and complexity of the method.

Third, the current implementation relies on a simple constraint-solving procedure that works only when there are no cyclic constraints. The procedure follows one or more finite paths in a constraint graph (which is acyclic) and sequentially updates each graphic object encountered on the way. An obvious shortcoming is that legitimate design alternatives may well be dismissed. As a simple (yet easily overlooked) example, currently two or more variable points are not allowed to be consecutive vertices of a line, because they would be implicitly constrained to each other and result in cyclic constraints. A problem of determining the location of adjacent variable points simultaneously (see Fig. 5) is indeed an interesting optimization problem, but cannot be solved by the aforesaid constraint-solving procedure. The problem is even harder if variable points are constrained to each other by user-specified constraints. As such, (geographic) constraint solving is a primary research agenda.

Lastly, at least from the user's perspective, the method should not be associated with any particular data model, although a raster model is employed here because it facilitates the immediate implementation of the method. Use of other data models would likely to cause significant computational difficulties in updating graphic objects. To see this, suppose that a vector model is used. Then, a spatially-varying condition may be represented by a subdivision of polygons, each having an associated value, or a polyhedral surface like a triangulated irregular network (TIN) if the map represents topographic elevation. Then, Operations 1 and 2 must be adapted to solve the shortest path problem on weighted planar polygonal subdivisions (Mitchell and Papadimitriou 1991) or on polyhedral surfaces (see, Bose et al. 2011, for a survey) which is less tractable than the classic shortest path problem on graphs. Likewise, Operation 4 would be further complicated, as it must extract circular, size-minimizing/maximizing, or other types of areas from such surfaces.

5 Conclusions

The paper introduced a method for constrained-based drawing in a digital geographic space, and discusses a data model and functionality essential for it. In this method, constraints may concern sizes of graphic objects or relationships between graphic objects that are dependent on spatially-varying conditions on or near the Earth's surface represented by single-factor maps. The method enables the user to edit graphic objects with geographic constraints incrementally and parametrically, while updating all affected graphic objects automatically. It was shown that the method can be implemented relatively straightforwardly using a raster model and a directed constraint graph, but not without limitations. These include inefficiency of the area-update operation, limited variety of size and relationship types, prohibition of cyclic constraints, and dependence on a particular spatial data model (i.e., raster). Nevertheless, it is hoped that the paper has highlighted a potential impact of the constraint-based drawing method on research and development of Computer-Aided GeoDesign in geographic information systems.

References

Aerts, J. C. J. H., Eisinger, E., Heuvelink, G. B. M., & Stewart, T. J. (2003). Using linear integer programming for multi-site land-use allocation. *Geographical Analysis,35*(2), 148–169.

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Englewood Cliffs, Upper Saddle River: Prentice Hall.

Batty, J. M., & Densham, P. J. (1996). Decision support, GIS, and urban planning. *Systemma Terra,5*(1), 72–76.

Betting, B., & Hoffmann, C. (2011) Geometric constraint solving in parametric computer-aided design. Journal Computing Information Science Engineering, *11*. doi:10.1007/1.3593408.

Bose, P., Maheshwari, A., Shu, C., & Wuhrer, S. (2011). A survey of geodesic paths on 3D surfaces. *Computational Geometry: Theory and Applications,44*, 486–498.

Church, R. L., Gerrard, R. A., Gilpin, M., & Sine, P. (2003). Constructing cell-based habitat patches useful in conservation planning. *Annals of the Association of American Geographers,93*(4), 814–827.

Cova, T. J., & Church, R. L. (2000). Contiguity constraints for single-region site search problems. *Geographical Analysis,32*(4), 306–329.

Digital Landscape Architecture. (2009–2014). url:http://www.digital-la.de/.

GeoDesign Summit. (2010–2014). url:http://www.geodesignsummit.com/.

Gleicher, M., & Witkin, A. (1994). Drawing with constraints. *The Visual Computer: International Journal of Computer Graphics,11*(1), 39–51.

Haverkort, H., Toma, L., & Zhuang, Y. (2009). Computing visibility on terrains in external memory. *ACM Journal of Experimental Algorithmics,13*(5), 1–23.

Jankowski, P. (1995). Integrating GIS and multiple criteria decision making methods. *International Journal of Geographical Information Systems,9*(3), 252–273.

Jenson, S. K., & Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing,54*(11), 1593–1600.

Kondo, K. (1990). PIGMOD: parametric and interactive geometric modeller for mechanical design. *Computer-Aided Design,22*(10), 633–644.

Ligmann-Zielinska, A., Church, R., & Jankowski, P. (2008). Spatial optimization as a generative technique for sustainable multiobjective land-use allocation. *International Journal of Geographical Information Science,22*(6), 601–622.

McHarg, I. L. (1969). *Design with nature*. New York: American Museum of Natural History.

Mitchell, J., & Papadimitriou, C. (1991). The weighted region problem: Finding shortest paths through a weighted planar subdivision. *Journal of the ACM,38*(1), 18–73.

Nalle, D. J., Arthur, J. L., & Sessions, J. (2003). Designing compact and contiguous reserve networks with a hybrid heuristic algorithm. *Forest Science,48*(1), 59–68.

Owen, J. C. (1991). *Algebraic solution for geometry from dimensional constraints* (pp. 397–407). Austin: ACM Symposium on the Foundations of Solid Modeling.

Önal, H., & Wang, Y. (2008). A graph theory approach for designing conservation reserve networks with minimal fragmentation. *Networks,51*(2), 142–152.

Shirabe, T. (2005). A model of contiguity for spatial unit allocation. *Geographical Analysis, 37*(1), 2–16.

Shirabe, T. (2008). Minimum work paths in elevated networks. *Networks,52*(2), 88–97.

Shirabe, T. (2009). Map algebraic characterization of self-adapting neighborhoods. In: K. S. Hornsby, C. Claramunt, M. Denis & G. Ligozat (Eds.), Spatial Information Theory: Cognitive and Computational Foundations. Lecture Notes in Computer Science. In *Proceedings of the Ninth International Conference on Spatial Information Theory: COSIT'09* (Vol. 5756, pp. 280–294). Berlin: Springer.

Shirabe, T. (2011). A heuristic for the maximum value region problem in raster space. *International Journal of Geographic Information Science,25*(7), 1097–1116.

Spatial concepts in GIS and design. (2008). url:http://ncgia.ucsb.edu/projects/scdg/.

Tomlin, C. D. (1990). *Geographic information systems and cartographic modeling*. Englewood Cliffs: Prentice-Hall.

Williams, J. C. (2002). A zero-one programming model for contiguous land acquisition. *Geographical Analysis,34*(4), 330–349.

Woeginger, G. J. (1992). Computing maximum valued regions. *Acta Cybern,10*(4), 303–315.

Xiao, N. (2006). An evolutionary algorithm for site search problems. *Geographical Analysis,38* (3), 227–247.

Voluminator—Approximating the Volume of 3D Buildings to Overcome Topological Errors

Horst Steuer, Thomas Machl, Maximilian Sindram, Lukas Liebel and Thomas H. Kolbe

Abstract Many research fields analysing urban space depend on 3D city models. However, 3D city models still are often of a low geometric quality. Due to topological errors it is not possible to compute volumetric information for many buildings in real world datasets using analytical approaches. Since this volumetric information is important for many applications we present a method to approximate the volume of building models overcoming topological errors. The method is based on a voxelisation and a generalisation of the point-in-polygon test to three dimensions. We show in extensive tests that the method produces accurate results and is able to cope with different types of errors. Beyond the computation of volumes, the proposed approach potentially has a high impact for numerous applications like healing of building models, indoor routing or model transformation.

Keywords Building models · Topological errors · Approximation · Voxelisation · Robust algorithm

H. Steuer (✉) · T. Machl · M. Sindram · L. Liebel · T.H. Kolbe
Technical University Munich, Munich, Germany
e-mail: steuer@tum.de

T. Machl
e-mail: thomas.machl@tum.de

M. Sindram
e-mail: maximilian.sindram@tum.de

L. Liebel
e-mail: lukas.liebel@tum.de

T.H. Kolbe
e-mail: thomas.kolbe@tum.de

F. Bação et al. (eds.), *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, DOI 10.1007/978-3-319-16787-9_20

1 Introduction

Within the last decade a paradigm shift in modelling geographic data has occurred: where formerly 2D data have been the predominant representation of geographic objects today more and more 3D datasets become available [even though there has been work on digital 3D geographic information as far back as 1971 (Brooks and Pinzke 1971)]. In the future this process has the potential to speed up even more because of the fast progress of consumer tailored technologies like augmented reality and its applications demanding large spatial datasets. Nevertheless, the problem of generating 3D models of geographic objects is not solved sufficiently yet. For example, Musialski et al. (2013) state for the case of urban 3D reconstruction that "in practice, full automation turns out to be hard to achieve" and that in order to gain high quality models, user interaction often is necessary. Furthermore they state that due to the complex scenes "it is often difficult to acquire coherent and complete data of urban environments". In view of that, it is no surprise that many current city models contain faulty building models (cf. Rottensteiner et al. 2012; Zhao et al. 2014 and Sect. 3). In order to meet the demands of a multitude of recording procedures, the data models of GIS typically are lenient. The popular standard CityGML (Kolbe et al. 2005) for modelling urban regions allows for buildings to be modelled either as GML solids or GML multisurface. Especially in the case of multiple surfaces it is possible that these surfaces are not touching due to geometric inaccuracies or that e.g., a wall is not represented at all by a GML surface member and therefore the building model is not a valid, closed solid (see Fig. 1).

In contrast, current analysis tools [e.g., FME[1] or GeOxygene 3D (Brasebin 2009)] in the field of geographic information systems have to rely on the topological correctness of given 3D models. For example in order to compute the volume of a 3D object, it has to be either represented by a volumetric model or a 'waterproof' boundary representation.

Given that there are applications [for example the city-wide energy estimation presented in Kaden and Kolbe (2013)] requiring a huge amount of highly accurate data there is a need to bridge the gap between the faulty geometries and the unforgiving analysis tools. A popular approach is to repair the given geometries by applying topological reasoning (Alama et al. 2013; Bogdahn and Coors 2010; Zhao et al. 2014), though due to the high complexity of the investigated objects, this problem is not solved yet: e.g., missing surfaces of a solid may be detected by geometry validation tools (Wagner et al. 2013), but none of the approaches mentioned above is able to reconstruct these missing surfaces in a fully automatic way. More specifically, Boeters (2013) states that working with the city model of Rotterdam 3D, the "largest problem is that no walls are modelled between neighbouring buildings causing 95 % of the buildings to not be solids because of missing faces in the shells" and that the approach of Wagner et al. (2013) can repair only "about 60 % of the dataset."

[1] http://www.safe.com/.

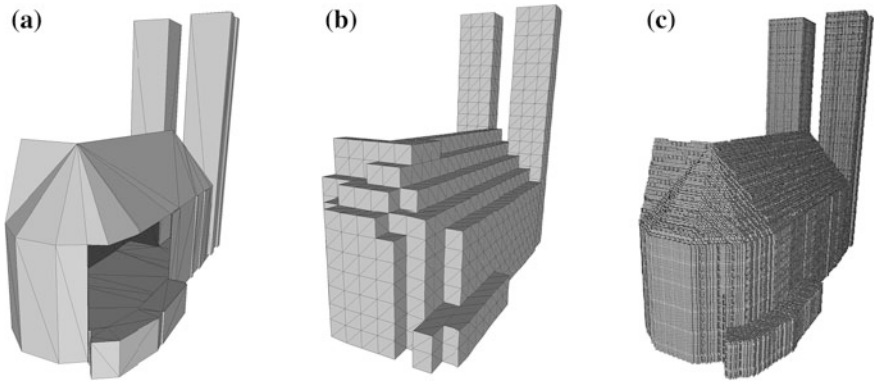


Fig. 1 Volumetric approximation of a building from an erroneous polygon model: Despite a large missing wall surface in a polygonal model of a church (**a**), using our approach volumetric models (**b**) and (**c**) with arbitrary resolutions can be derived. **a** Polygon model. **b** Voxel model with resolution 5 m. **c** Voxel model with resolution 0.5 m

Other approaches aim at deducing information from faulty geometry by creating robust algorithms. Often these algorithms are designed for coping just with numerical errors (Zhang et al. 1995) and therefore are not suitable for many of the errors typically occurring in geographic models. Biljecki et al. (2014) analyse the impact of geometrical (not topological) errors on the result of an analytical volume computation. They state that even small geometric errors in the building models may lead to big errors in the computed volumes.

We present a method to robustly compute the volume of 3D buildings that can cope with topological errors. Our approach works directly on building models containing topological errors without repairing the models first. Overcoming these errors, volumetric information can be deduced from the building models. Additionally, our approach can be utilized for the detection and localization of topological errors (Sect. 4).

In the following section we introduce the proposed algorithm. Section 3 gives an extensive evaluation of the algorithm analysing achievable accuracies and computation speeds using three different datasets. We conclude this work in Sect. 5 with a summary of the proposed method and an outlook to future refinements as well as potential fields of application.

2 Calculation of Volumes

This section describes our approach to the calculation of volumes for building models with topological errors. We assume that a building is described by its outer shell and aim at the determination of the volume inside this outer shell. The approach is suited to calculate the volumes of single rooms of a building or different volumetric objects described by their outer and inner shells.

2.1 Robust Voxelisation

The basic idea of our approach is to discretise the space of a buildings bounding box into a three-dimensional grid V. For each of the cells v (in the following denoted as *voxel*) of this grid, we try to decide if this cell is inside or outside the building.

The test, which leads to this decision, is based on the point-in-polygon test as described by Sutherland et al. (1974). In the 2D case we can decide if a point P lies inside the polygon by counting the number of intersections of a ray starting at P and going into an arbitrary direction. As is illustrated in Fig. 2a the number of intersections is uneven if P is inside the polygon and even or zero otherwise. This point-in-polygon test requires that the polygon does not intersect itself.

If the geometry of the polygon is defective, e.g., the linear ring describing the outer border of the polygon is not closed because of a missing line segment as depicted in Fig. 2b, a single ray may lead to a wrong decision. The probability for a correct decision can be increased by incorporating multiple rays into the decision process. As illustrated in Fig. 2b, three of the four rays intersect an uneven number of lines, only one does not intersect the model. Because the majority of these four rays indicate that P_1 is inside the polygon, the overall decision is likewise.

This method can be generalized to the three dimensional space (c.f. Nooruddin and Turk 2003): Here, it can be tested if a point lies inside a volumetric geometry. In the case of a building model we assume that the volumetric object is represented by a set of polygons B denoting the outer shell. By testing several rays $r_i = P + \lambda\overrightarrow{dir_i}$ with $\lambda \in \mathbb{R}+$ and $\overrightarrow{dir_i} \in \mathbb{R}^3$ a directional vector starting from a point $P \in \mathbb{R}^3$ for intersection with B (i.e., computing the sets $r_i \cap B$, with $r_i, i \in 1, \ldots n$

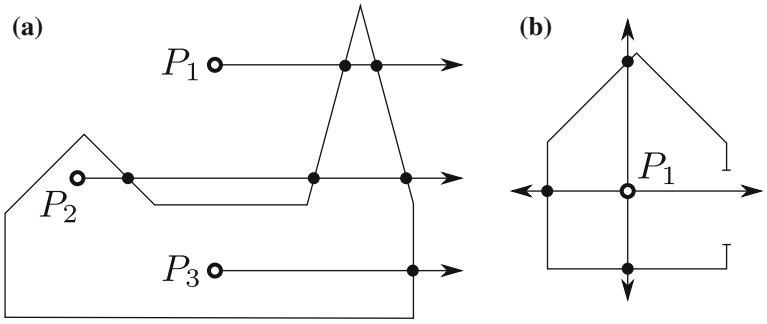


Fig. 2 Point-in-polygon test: **a** Shooting a ray into an arbitrary direction results in an even number of intersections for initial points (e.g., P_1) outside the topological correct polygon and an uneven number for initial points (P_2 and P_3) inside the topological correct polygon. **b** In the case of a defective polygon, the probability for a correct decision can be increased by incorporating multiple rays into the decision process using a majority vote. **a** Point-in-polygon test for a topological correct polygon. **b** Point-in-polygon test for a defective polygon

the n rays starting at the centre of a cell v), we can decide if a point P lies inside the volumetric object, even if the set of polygons B does not yield a watertight volume. In order to determine the set $A \subseteq V$ of voxels on the inside of outer shell we test the centre of each voxel v in the constructed grid V by constructing six evenly distributed rays (in positive and negative direction along each coordinate axis) in this way. A result of this approach is depicted in Fig. 1b, c shows sets of voxels which are marked as lying inside the erroneous building model in (a).

In order to compute the volume of a building, the following sum has to be calculated:

$$\sum_{v \in V} c\mathbf{1}_A(v)$$

where c is a constant value depending on the volume of a single voxel defined by the resolution and $\mathbf{1}_A(v)$ an indicator function

$$\mathbf{1}_A(v) := \begin{cases} 1 & \text{if } votes(v) > \frac{n}{2}, \\ 0 & \text{else.} \end{cases}$$

and

$$votes(v) = \sum_{i=1}^{n} \begin{cases} 1 & \text{if } \|(r_i \cap B)\| \text{is uneven and finite,} \\ 0 & \text{else.} \end{cases}$$

$\|(r_i \cap B)\|$ denotes the number of intersections of one ray r_i with the set of polygons B. Note that in the special case of a ray r_i lying in the plane of a polygon of B this cardinality of intersections may be infinite and therefore not determinable even or uneven.

This sum can be evaluated without keeping information about the complete set of voxels in memory, therefore the algorithm is not memory constricted.

By choosing different voxel sizes, the accuracy of the results and the computation time can be affected. We analyse these effects in Sect. 3.2. The algorithm can effectively overcome topological errors like missing surfaces, models with cracks between surfaces or surfaces which are included multiple times in a dataset. Using real world as well as artificial data, we analyse to which extent the algorithm is robust to these errors in Sect. 3.

3 Experiments

We analysed the properties of our approach with regard to accuracy and runtime. In the following section we give a short overview over the datasets we used in our experiments. In Sect. 3.2 we show the results of an experiment analysing the

accuracy of the presented algorithm using topological correct building models both from synthetic as well as real world datasets. More specifically, we analyse the impact of the voxel size on the accuracy as well as differences between the approaches. In Sect. 3.3 we test the approaches for robustness against topological errors. Here we use topological correct data as a ground truth and add errors to the models. Using real world data with real errors we analyse the algorithm in Sect. 3.4 qualitatively. In Sect. 3.5 we analyse the impact of voxel sizes on the runtime.

3.1 Datasets

We tested the presented approach using three different datasets. Two datasets were given as LOD2 CityGML, one of those representing a real inner city region while the other one is a synthetic set of buildings. The third dataset contains buildings represented only by multi-patches without semantic information and represents a real world village. In the following we give a short description of the different datasets.

3.1.1 Synthetic Dataset

In comparison to real world data, synthetic data have the advantage of allowing for arbitrary geometric and topological accuracy. In our case, we were interested in an error free dataset which allows for a computation of the volume using an analytical approach. Using this accurate volume as a ground truth we tested the accuracy of our approach for optimal geometries (cf. Sect. 3.2).

As a synthetic dataset we chose the dataset LOD2-F1 from *Random3Dcity*, which is available at http://3dgeoinfo.bk.tudelft.nl/biljecki/Random3Dcity.html. It contains 1600 buildings represented as LoD 2 models. The dataset contains only relatively small, detached buildings. In comparison to the real world datasets, the synthetic dataset consists of geometrically relative homogeneous building models. An excerpt of this dataset is depicted in Fig. 3a. The generation of these models is described in Biljecki et al. (2014). In the following we denote this dataset as *Synth*.

Fig. 3 Excerpts of the three datasets used in the experiments. **a** Synth. **b** Munich. **c** Fischach

3.1.2 Munich

For testing our approach we also used a real world dataset thankfully provided by the City of Munich[2] representing an excerpt of the city. This dataset denoted as *Munich* includes 3178 building objects in a region of roughly $2.6\,\text{km} * 0.9\,\text{km} = 2.34\,\text{km}^2$. The buildings are represented by a LoD2 geometry (i.e., outer hull including the roof structure) using multi surfaces.

The building models have been generated in an automatic workflow by applying a spatial join to a digital elevation model stemming from photogrammetric point matching and footprints of buildings. In this dataset many GML Polygons are not planar. Since our algorithm assumes the planarity of polygons, we disaggregated these polygons into triangles in a preprocessing step. An excerpt of this dataset is depicted in Fig. 3b.

Of the LoD2 models in this dataset $263 \hat{=} 8\,\%$ constitute a topological correct, water tight model. Despite this fact, we were able to generate 2671 closed building models using the *SolidBuilder* transformer of FME which we used as a basis for the datasets described in the following. For 2408 of these generated closed building models we observed unused surfaces.

In order to test our approach for robustness against topological errors, we chose the subset of closed building models from this dataset as ground truth data and added different types of artificial errors. For each type of error, we created two datasets: (a) one randomly chosen surface per building is affected by the error and (b) four randomly chosen surfaces are affected by this error. In this way we created the following datasets:

- *Rotation-1* and *Rotation-4*: we rotated the chosen surfaces into randomly chosen directions by randomly chosen angles of up to 10°.
- *Doubling-1* and *Doubling-4*: we duplicated the chosen surfaces.
- *Deletion-1* and *Deletion-4*: we deleted the chosen surfaces.
- *Translation-1* and *Translation-4*: we translated the chosen surfaces by randomly chosen distances of up to 1 m along the three coordinate system axis.
- *Orientation-1* and *Orientation-4*: we reversed the orientation of the chosen surfaces, i.e., we reversed their normal vector by changing the order of its vertices.

3.1.3 Fischach

The second real world dataset represents Fischach, a village in the west of Munich. It contains 189 building models in a region of roughly $950\,\text{m} * 590\,\text{m} = 0.5605\,\text{km}^2$. In contrast to the other two datasets, the outer hull of the buildings of the dataset *Fischach* consist of a very high number of triangles. This is due to the simple

[2]The City of Munich holds the copyright for this dataset.

generation process of the models. The buildings were generated by extruding building ground plans between a digital terrain model and a digital surface model both given as triangulated irregular networks. The digital elevation models originated from aerial LIDAR measurements. The vertices of both digital elevation models were regridded using an even distribution with a resolution of one vertex per 0.5 m. Since the building models were not post processed in any way, the ground and roof surfaces are identical in resolution to the used digital elevation models. The generation process guarantees topological closed building models, while the models do contain typical geometric errors stemming e.g., from vegetation. An excerpt of this dataset can be seen in Fig. 3c.

3.2 Accuracy for Topological Correct Data

In order to determine if the proposed approach can compute volumes with a sufficient accuracy, we tested the algorithms using topological correct data and compared our results with analytically calculated volume values (using FME) as a ground truth. In this experiment we used the topological correct buildings from *Munich*. Furthermore, we used the datasets *Fischach* and *Synth* since they are free of topological errors due to the methods they have been acquired with.

We tested the algorithm with four different voxel sizes: $\{2\,\text{m}, 1\,\text{m}, 0.5\,\text{m}, 0.25\,\text{m}\}$. In Fig. 4 you can see probability density functions for the four voxels sizes using a

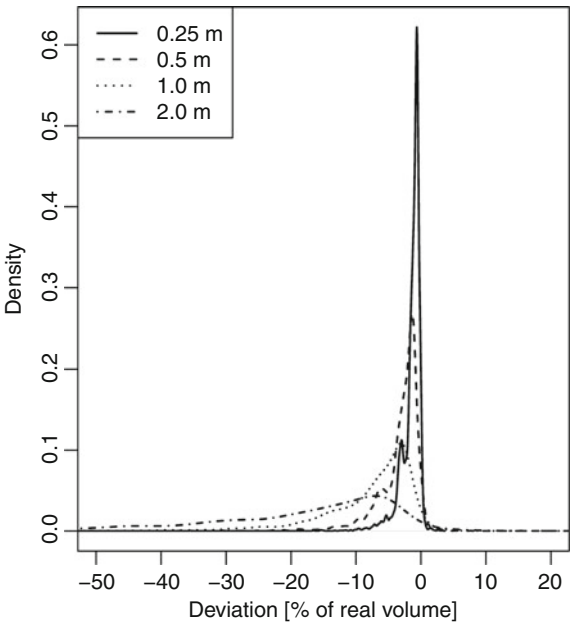


Fig. 4 Impact of resolution on accuracy: probability density functions for different resolutions shows that the algorithm tends to undervalue the volume. The relative error clearly is reduced when using higher resolutions

combination of buildings from all three datasets (a total of 4460 building models). The x-axis represents the relative deviation of the computed volumes to the ground truth data. The ordinate shows the density of the relative deviations. As was to be expected, large voxel sizes lead to less accuracy than small voxel sizes. A voxel size of 0.25 m yields a very low deviation. As an additional result of this experiment, we conclude that the algorithm tends to undervalue the ground truth values.

3.3 Robustness Against Topological Errors

The main reason for the research conducted was to find a method to calculate volumes of topological erroneous building models. In this experiment we tested the approach for robustness against different kinds of topological errors. As a ground truth we used the topological correct subset of buildings from *Munich* and computed the volumes of buildings using an analytical approach. We added errors to the building models as described in Sect. 3.1.2 resulting in 10 datasets containing different kinds and numbers of errors. The total number of buildings in these 10 datasets is 26,710.

For each building model of these 10 datasets we computed the volume values using the approach described in Sect. 2.1 using a resolution of 0.5 and 1 m.

In Fig. 5 you can see selected results for four of these datasets using a resolution of 1 m. Additional graphs for the remaining three types of errors and for a resolution of 0.5 m are depicted in the appendix in Figs. 10 and 11, respectively. On the x-axis the volume values we assume as ground truth are shown while the ordinate shows the values computed using the presented approach. The regressions show a clear linear correlation and a low deviation from the model. This is especially the case for the datasets with just one error per building. Here the slope of the regression function differs only by a small amount from the optimal slope of 1. The coefficient of determination indicates a good fit for the linear model. For the datasets containing four errors per building it is noticeable, that only the error types *Deletion* and *Doubling* lead to a perceivable increase of dispersion.

The observation that the algorithm tends to slightly underestimate the volume values as described in Sect. 3.2 can be confirmed by this experiment.

3.4 Testing with Real Errors

Due to the lack of a ground truth it is not possible to conduct a quantitative evaluation of the robustness against real world errors. Instead, we chose a qualitative evaluation by visualizing the computed voxel models and comparing them to the original, erroneous building models. For this experiment, we chose the erroneous buildings from *Munich*. In Fig. 6 exemplary visualisations of two buildings

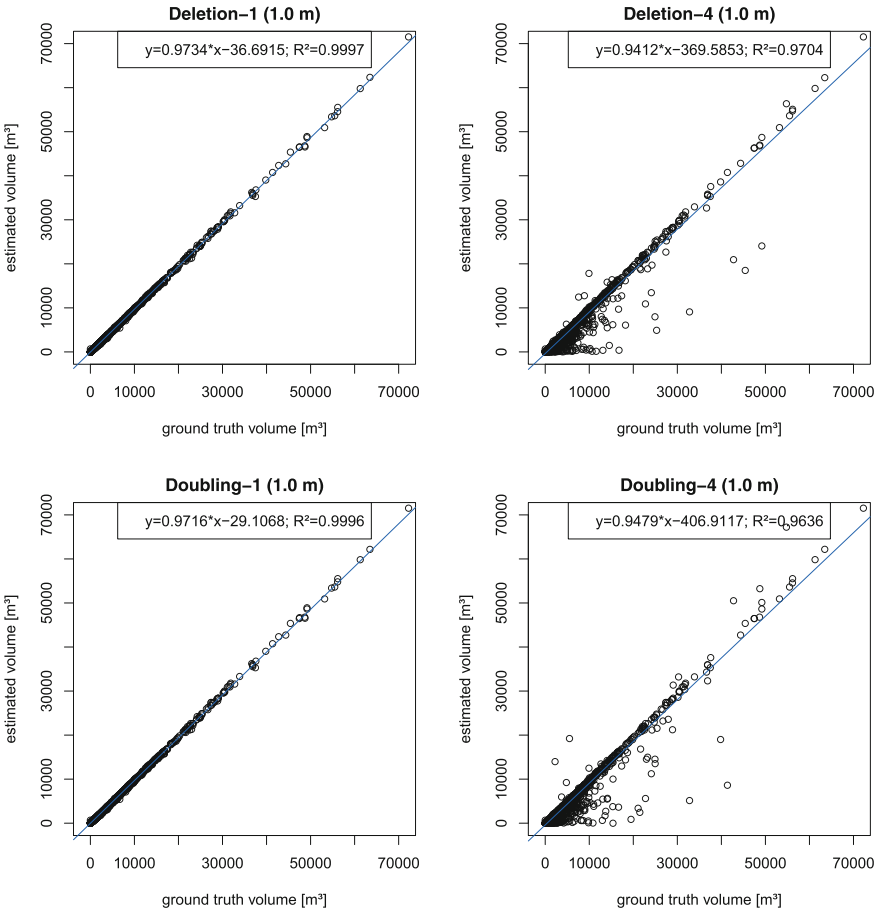


Fig. 5 Impact of topological errors on the accuracy of the results: exemplary regressions for the error types *Deletion* and *Doubling* using a voxel size of 1 m

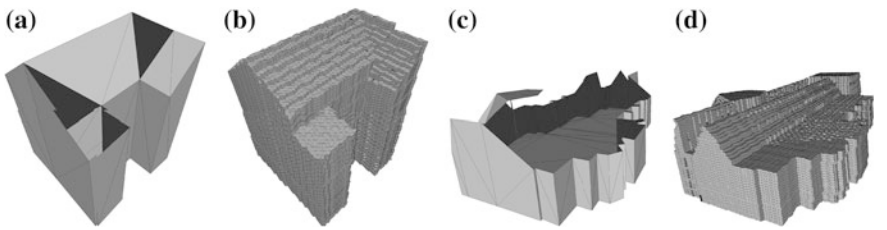


Fig. 6 Voxelisations of real world data. **a** and **c** Depict building models from *Munich* with missing roof surfaces. Despite these errors, the proposed algorithm generates reasonable voxel models (**b**) and (**d**) (resolution: 0.5 m)

models and the resulting voxel models are shown. Despite missing roof surfaces the results seem to be reasonable. The voxel models do not have gaps and approximate the roof structure well. In some cases there were so many roof surfaces missing, that it is difficult for a manual operator to interpret the roof structure. For these cases the algorithm at least was able to generate a voxel model accurately depicting the storeys under the roof and therefore to approximate the volume of the building model.

We added additional visualisations of some of the analysed building models at the end of the paper in Fig. 12.

3.5 Runtime Evaluation

In order to test the impact of the resolution on the runtime, we used a similar setup as described in Sect. 3.2. As input data, we used the topological correct buildings from *Munich* and the complete datasets *Fischach* and *Synth*. We computed the volumetric values of all buildings using resolutions of $\{2\,\text{m}, 1\,\text{m}, 0.5\,\text{m}, 0.25\,\text{m}\}$. The computations were conducted on a desktop PC with two Xeon E5-2609 at 2.4 GHz on 6 CPU-cores. The presented runtimes denote the sum of computation times over all used CPU-cores.

In Fig. 7 the runtimes for each of the three datasets is shown against the voxel sizes. Since the ordinate has a logarithmic scale, the runtimes show an exponential growth with increasing resolution (i.e., decreasing voxel sizes).

The relatively small variance in runtimes for the dataset *Synth* reflects the relative low geometric variance of the buildings when compared to the two real world datasets.

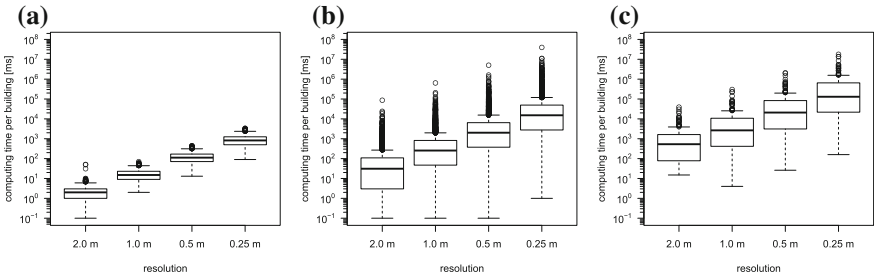


Fig. 7 Runtime analyses: The three plots show the runtime for the three datasets against the resolution. The bottom of the boxes denote the first quartile, the top of the boxes is the third quartile, while the band inside the box denotes the median. The circles above the box were treated as outliers. The scale of the ordinate is logarithmic. **a** Synth. **b** Munich. **c** Fischach

4 Inference of Quality Information

The proposed method allows for additional applications. In this section we outline a basic technique for localizing errors in the polygon models based on the robust voxelisation.

For each voxel inside a watertight model, all rays should have an uneven number of intersections with the polygons of the model. If this is not the case for a model, we can assume that it is not watertight or that it has different errors. By visualizing the number of rays voting for the insideness for each voxel through colour encoding, the algorithm can provide the user with hints, as to where problems in the original polygon model exist. A demonstration of this is illustrated in Fig. 8. In Fig. 8a voxels with a vote of six are coloured grey, with a vote of five bright green, with a vote of four dark green and so on. Comparing the coloured voxel model with the initial model in Fig. 8b shows that missing surfaces and clefts between surfaces result in coloured voxels.

It is possible to localize potential errors in the initial model with a higher accuracy. Instead of analysing only the number of votes $votes(v)$, we can analyse for each voxel, how many intersections the different rays r_i have with the set of boundary surfaces B. This can be described by a feature vector

$$\mathbf{f}_{\mathrm{v}} = \begin{pmatrix} \|r_1 \cap B\| \\ \|r_2 \cap B\| \\ \cdots \\ \|r_n \cap B\| \end{pmatrix}.$$

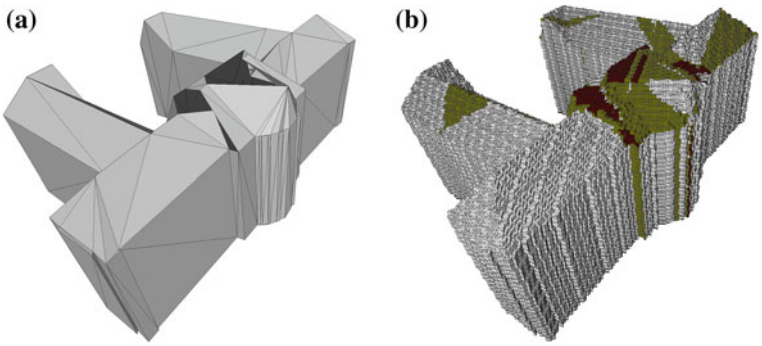


Fig. 8 Finding errors in the polygonal model: **a** colour-encoding the number of ray-polygon intersections $\|(R_v \cap B)\|$, **b** highlights areas with low support of the original polygonal model (**a**). **a** Original model. **b** Colour encoded voxel model

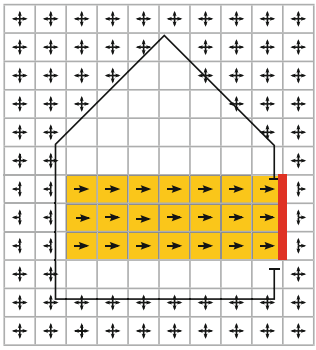


Fig. 9 Localizing errors in a polygonal model (2D projection). The *arrows* in each voxel visualize the direction of the rays which vote for the voxel being outside the polygonal model. The *yellow* voxels are a group of voxels with identical feature vectors. The border (*red line*) of this group of voxels in the direction of the ray which indicated that the voxels are outside the building coincides roughly with the missing polygon in the model and might help to identify the error or fill holes

By computing sets of connected voxels where each voxel has the same feature vector we can get groups of voxels which potentially are influenced by the same error in the original model. Using $\mathbf{f}_v$ we can find out in which direction relative to the voxels the error occurred (cf. Fig. 9). By identifying the surface of the voxel-group lying in the direction determined by $\mathbf{f}_v$ we can find a possible location of the error. This way, a manual operator may be supported in finding errors in polygonal building models.

5 Conclusion and Outlook

In this contribution we presented a method to approximate the volume of building models overcoming topological errors. Based on an extension of the point-in-polygon test to 3D space we devised a simple but effective algorithm. Extensive testing of the algorithm using both artificial and real world data has shown that the results indeed approximate the ground truth values while tending to a slight underestimation. We analysed the performance concerning the accuracy of the algorithm for five different error classes with two different levels of severity and concluded that the proposed algorithm has a high robustness against all five types of errors. While the impact of rotated and translated boundary surfaces as well as a changing of orientation of these surfaces have a very uniform impact on the accuracy of the results, the error types *Deletion* and *Doubling* of surfaces result in a higher dispersion of errors.

The implementation of the proposed approach has not yet been optimised for computation speed. A first improvement would be to utilize spatial indices for the intersection tests. Since the evaluation of the presented indicator function $\mathbf{1}_A(v)$ can be performed independently for each voxel, the algorithm is a candidate for a parallel processing scheme. Another interesting extension of the algorithm would be to use an spatial subdivision schema like an Octree (Meagher 1982) instead of the regular grid employed in this approach. Since the decision if a cell is inside a building model is uncertain, it is not trivial to determine when a cell should be subdivided.

The proposed method has a potential for several extensions and applications to additional problems. As outlined in the last section, it is possible to achieve a rough localisation of topological errors using our approach. Further research can consider what kind of topological errors can be localized and how high the precision and reliability of this localisation is. In a next step it would be interesting to analyse how this error detection and localisation may be utilised for a (semi-) automatic repair process.

Another potential application of this voxelisation is the problem of indoor routing and navigation. Here, several approaches (Khan et al. 2014; Lin et al. 2013; Steuer 2013) compute the space where a moving agent (human or non-human) may move without collisions in 3D space but have to rely on topological correct indoor models. A voxelisation overcoming topological errors may be helpful to compute this free space effectively.

Transformations of 3D building models from Industry Foundation Classes (IFC) to CityGML have been researched quite thoroughly. The opposite direction of a transformation from CityGML to IFC has not been solved successfully yet, since it is difficult to derive volumetric data (which is needed by IFC) from surface data as is provided by CityGML. The proposed method may be helpful as a basis for a spatial reasoner achieving this transformation.

Appendix

In this section we present additional material illustrating the impact of topological errors on the accuracy of the proposed method for (Figs. 10 and 11) as well as visualisations of erroneous building models and the resulting voxel models (Fig. 12).

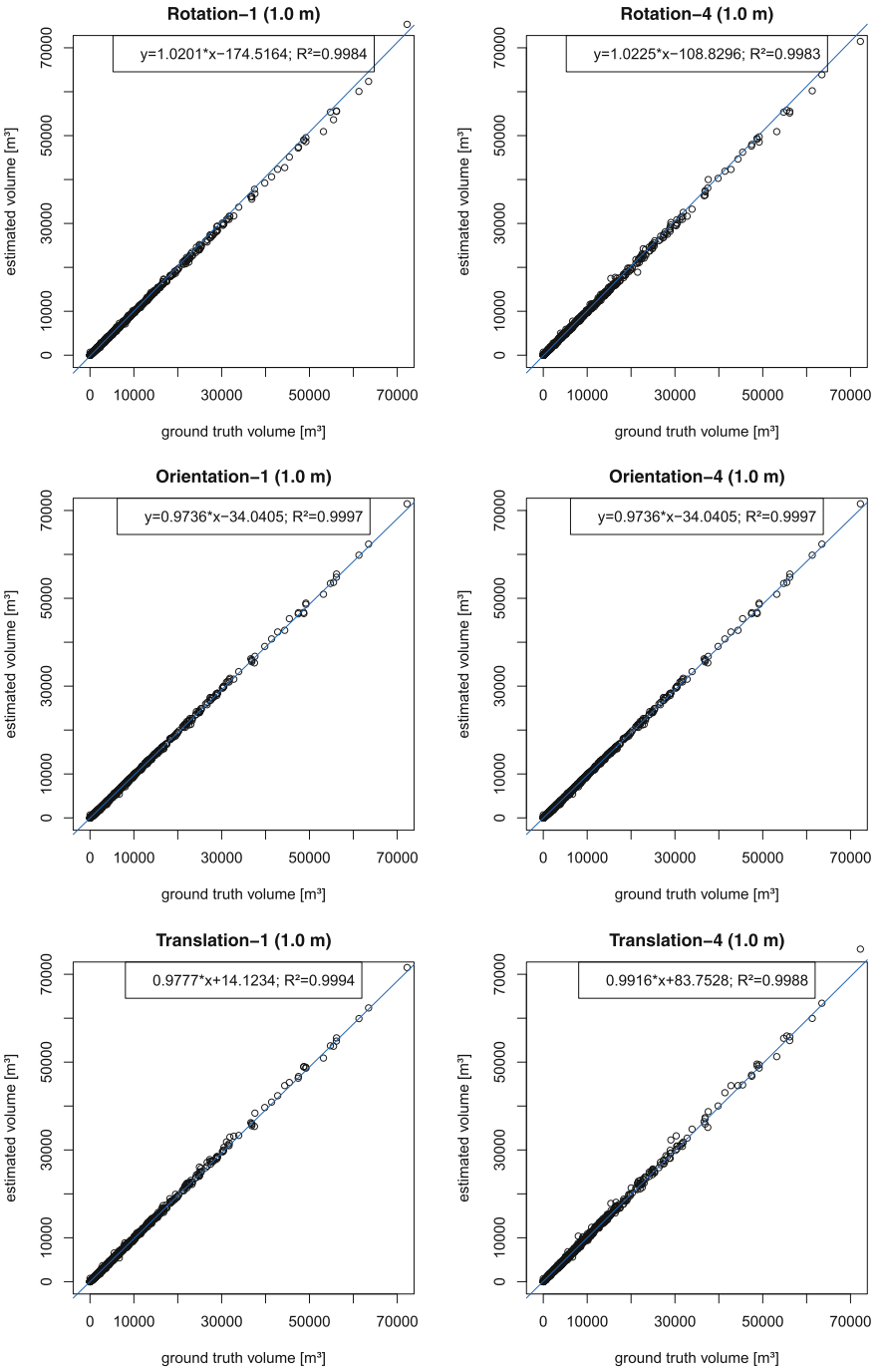


Fig. 10 Impact of topological errors on the accuracy of the results: exemplary regressions for the error types *Rotation*, *Orientation*, and *Translation* using a voxel size of 1 m

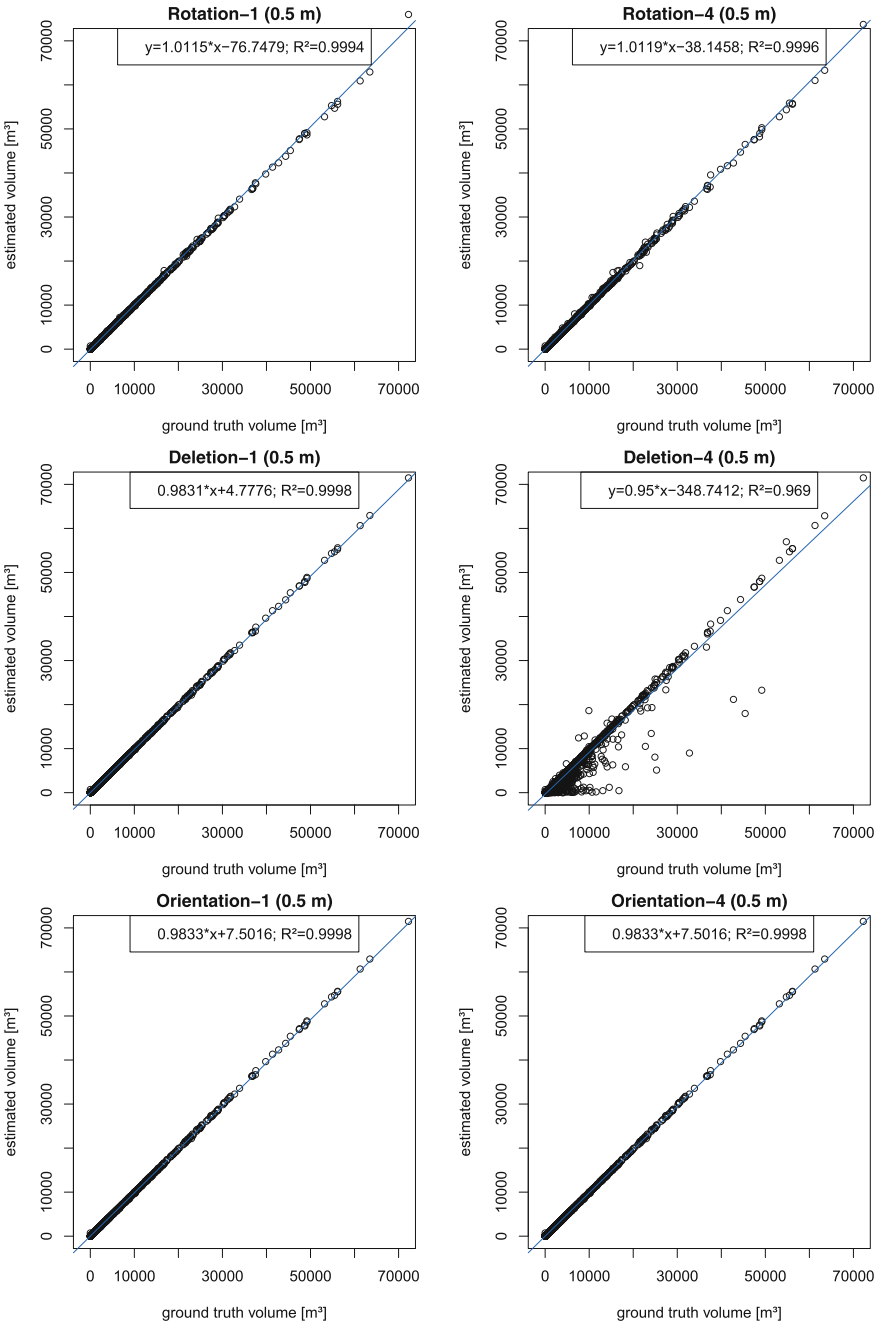


Fig. 11 Impact of topological errors on the accuracy of the results: exemplary regressions for all five types of errors using a voxel size of 0.5 m

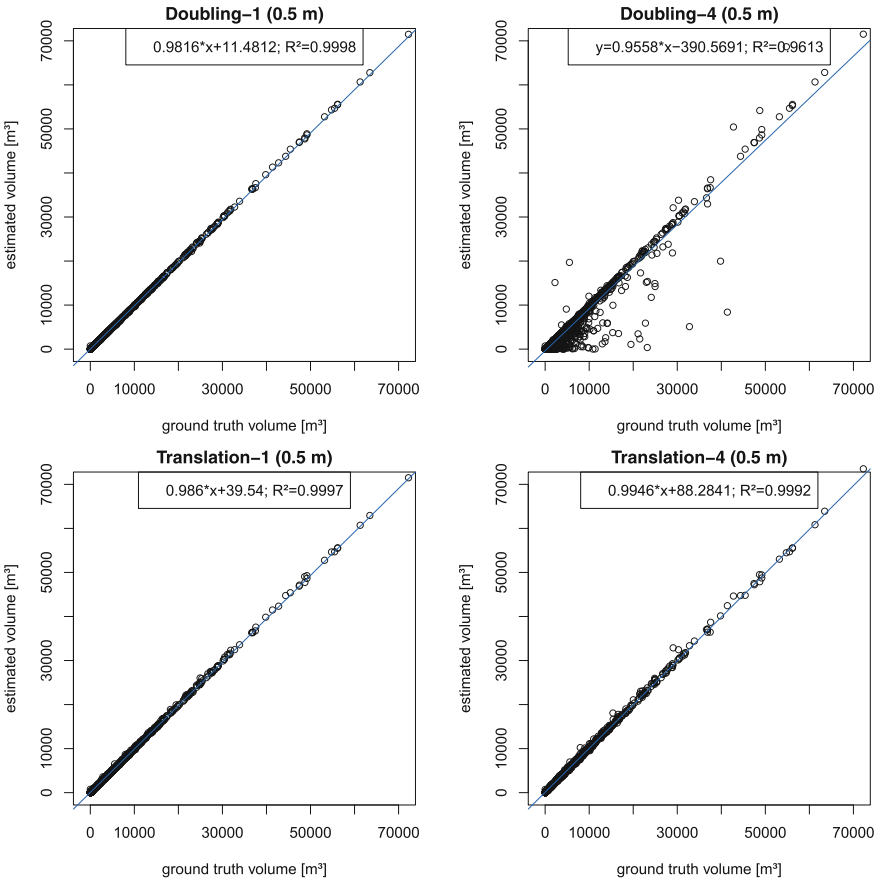


Fig. 11 (continued)

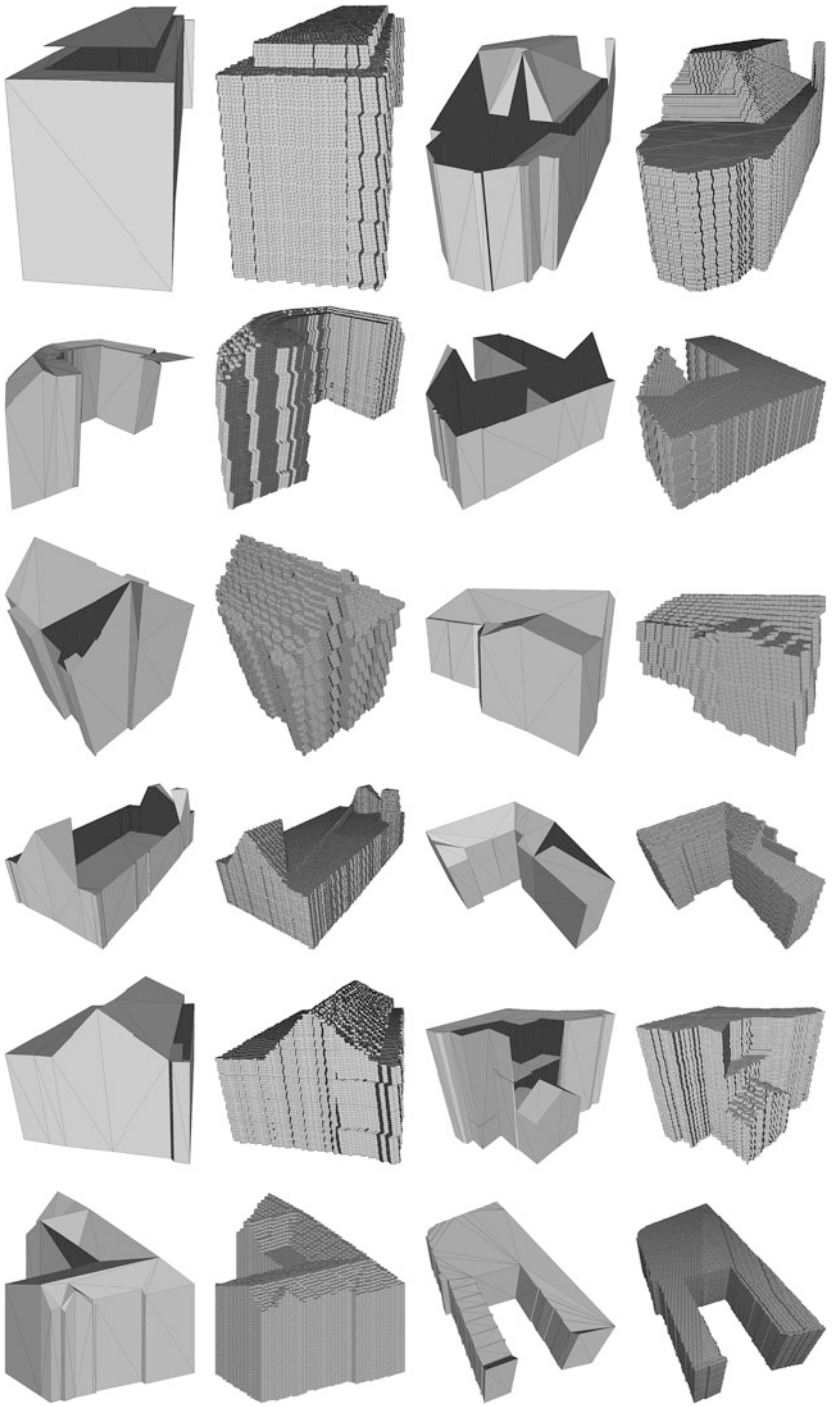


Fig. 12 Examples of voxelisations of buildings with real errors. The *first* and *third column* show the erroneous polygon model while the *second* and *fourth column* show voxelisation with a resolution of 0.5 m

References

Alama, N., Wagner, D., Wewetzer, M., von Falkenhausen, J., Coors, V., & Pries, M. (2013). Towards automatic validation and healing of citygml models for geometric and semantic consistency. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 1*(1), 1–6.

Biljecki, F., Ledoux, H., & Stoter, J. (2014). Error propagation in the computation of volumes in 3D city models with the monte carlo method. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS/IGU Joint International Conference on Geospatial Theory, Processing, Modelling and Applications, Toronto, Canada*.

Boeters, R. (2013). *Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination*. Ph.D. thesis, Master's thesis Section GIS Technology, Delft University of Technology. http://www.gdmc.nl/publications/2013/Automatic_enhancement_CityGML.pdf

Bogdahn, J., & Coors, V. (2010). Towards an automated healing of 3D urban models. In *Proceedings of international conference on 3D geoinformation. International archives of photogrammetry, remote sensing and spatial information science* (Vol. 38, p. 4). Citeseer.

Brasebin, M. (2009). Geoxygène: An open 3D framework for the development of geographic applications. In *Proceedings of AGILE*.

Brooks, W. D., & Pinzke, K. G. (1971). A computer program for three-dimensional presentation of geographic data. *Cartographica: The International Journal for Geographic Information and Geovisualization, 8*(2), 110–125.

Kaden, R., & Kolbe, T. H. (2013). City-wide total energy demand estimation of buildings using semantic 3D city models and statistical data. In *Proceedings of the 8th International 3D GeoInfo Conference* (Vol. II-2/W1).

Khan, A. A., Donaubauer, A., & Kolbe, T. H. (2014). A multi-step transformation process for automatically generating indoor routing graphs from existing semantic 3D building models. In *Proceedings of the 9th 3D GeoInfo Conference*.

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). Citygml: Interoperable access to 3D city models. In *Geo-information for disaster management* (pp. 883–899). Berlin: Springer.

Lin, Y.-H., Liu, Y.-S., Gao, G., Han, X.-G., Lai, C.-Y., & Gu, M. (2013). The IFC-based path planning for 3D indoor spaces. *Advanced Engineering Informatics, 27*(2), 189–205.

Meagher, D. (1982). Geometric modeling using octree encoding. *Computer Graphics and Image Processing, 19*(2), 129–147.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Gool, L., & Purgathofer, W. (2013). A survey of urban reconstruction. In *Computer graphics forum* (Vol. 32, pp. 146–177). Wiley Online Library.

Nooruddin, F. S., & Turk, G. (2003). Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics, 9*(2), 191–205.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., et al. (2012). The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I-3*, 293–298.

Steuer, H. (2013). High precision 3D indoor routing on reduced visibility graphs. In *Progress in location-based services* (pp. 265–275). Berlin, Heidelberg: Springer.

Sutherland, I. E., Sproull, R. F., & Schumacker, R. A. (1974). A characterization of ten hidden-surface algorithms. *ACM Computing Surveys (CSUR), 6*(1), 1–55.

Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., & Coors, V. (2013). Geometric-semantical consistency validation of citygml models. In *Progress and new trends in 3D geoinformation sciences* (pp. 171–192). Berlin: Springer.

Zhang, Z., Deriche, R., Faugeras, O., & Luong, Q.-T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence, 78*(1), 87–119.

Zhao, J., Stoter, J., & Ledoux, H. (2014). A framework for the automatic geometric repair of citygml models. In *Cartography from pole to pole* (pp. 187–202). Berlin: Springer.