Lecture Notes
in Geoinformation and Cartography

LNG&C

Stan Geertman
Wolfgang Reinhardt
Fred Toppen *Editors*

Advancing Geoinformation Science for a Changing World

Springer

Lecture Notes in Geoinformation and Cartography

S.C.M. Geertman • W.P. Reinhardt
F.J. Toppen
Editors

Advancing Geoinformation Science for a Changing World

Springer

Editors
Stan Geertman
Faculty of Geosciences
Utrecht University
Heidelberglaan 2
3584 CS Utrecht,
The Netherlands
s.geertman@geo.uu.nl

Wolfgang Reinhardt
AGIS / Faculty of Computer Science
University of the Bundeswehr Munich
Werner-Heisenberg-Weg 39
85577 Neubiberg
Germany
Wolfgang.Reinhardt@unibw.de

Fred Toppen
Faculty of Geosciences
Utrecht University
Heidelberglaan 2
3584 CS Utrecht,
The Netherlands
f.toppen@geo.uu.nl

ISSN 1863-2246 e-ISSN 1863-2351
ISBN 978-3-642-19788-8 e-ISBN 978-3-642-19789-5
DOI 10.1007/978-3-642-19789-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011925152

Cover design: SPi Publisher Services

Printed on acid-free paper

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

Preface

The Association of Geographic Information Laboratories for Europe (AGILE) was established in early 1998 to promote academic teaching and research on GIS at the European level. Since then, the annual AGILE conference has gradually become the leading GIScience conference in Europe and provides a multidisciplinary forum for scientific knowledge production and dissemination. In that, it can be regarded a full successor of the preceding EGIS (European Conference on Geographical Information Systems) and JECC (Joint European Conferences and Collaboration) conferences, which dominated the European GI-conference scene during the nineties.

For the fifth consecutive year the AGILE conference promoted the edition of a book with the collection of scientific papers that were submitted as full-papers. Those papers went through a competitive review process. The 14th AGILE conference call for full-papers of original and unpublished fundamental scientific research resulted in 70 submissions, of which 26 were accepted for publication in this volume after a thorough selection and review process (acceptance rate of 37%).

The papers submitted to our Program Committee all can be considered to contribute to the 'Advancing Geoinformation Science for a Changing World', which is the overall title of this 14th AGILE conference. Therein we acknowledge that the pace of developments is increasing all the time and that our field of mutual interest – GIScience – can be considered a valuable player to cope in a proper way with these fast changing circumstances. We think that the papers included in this volume nicely reflect the contribution of GIScience to our ever-changing world.

The scientific papers published in this volume cover a wide diversity of GIScience related themes, including: spatial-temporal modeling and analysis; road network and mobility research; GeoSensor development and application; socio-spatial modeling and analysis; spatial data processing and structuring; and GI-information generation and dissemination.

Organizing the program of an international conference and editing a volume of scientific papers requires time, effort, and support. We would like to thank the authors for their high-quality contributions, which are invaluable for an edited volume. Moreover, we would like to thank the reviewers

for their difficult task to pick out those contributions that are really worthwhile for inclusion in such a book. Although this task always has a subjective part, by fulfilling the review process double-blind and demanding at least 3 reviews per submission we hope and expect to have overcome this subjectivity in sufficient manner. In addition we would like to thank the AGILE Council and Committees for their support.

We would also like to thank our sponsors (ESRI, the faculty of Geosciences, Utrecht University and the Royal Dutch Association of Geographers, KNAG) for their kind contribution to this conference. And last but for sure not least we would like to thank Springer Publishers for their willingness – already for the fifth time – to publish these contributions in their academic series *Springer Lecture Notes in Geoinformation and Cartography*.

Stan Geertman, Wolfgang Reinhardt, Fred Toppen (editors)

Utrecht/Munchen
February, 2011

Committees

Programme Committee

Programme Chair Stan Geertman
University of Utrecht (The Netherlands)

Programme Co-Chair Wolfgang Reinhardt
Universität der Bundeswehr München (Germany)

Programme Co-Chair Fred Toppen
University of Utrecht (The Netherlands)

Local Organizing Committee

Fred Toppen, University of Utrecht (The Netherlands) (chair)
JanJaap Harts, University of Utrecht (The Netherlands)
Henk Ottens, Royal Dutch Association of Geographers - KNAG, (The Netherlands)

Scientific Committee

Suchith Anand, University of Nottingham (United Kingdom)
Peter Atkinson, University of Southhampton (United Kingdom)
Fernando Bação, New University of Lisbon (Portugal)
Itzhak Benenson, Tel Aviv University (Israel)
Lars Bernard, TU Dresden (Germany)
Michela Bertolotto, University College Dublin (Ireland)
Ralf Bill, Universität Rostock (Germany)
Thomas Bittner, State University of New York at Buffalo (USA)
Lars Bodum, Aalborg University (Denmark)
Arnold Bregt, Wageningen University (The Netherlands)

Thomas Brinkhoff, Institute for Applied Photogrammetry and Geoinformation (Germany)
Christoph Brox, University of Muenster (Germany)
Gilberto Camara, National Institute for Space Research (Brazil)
Christophe Claramunt, Naval Academy Research Institute (France)
Arzu Cöltekin, University of Zürich (Switzerland)
Max Craglia, Joint Research Center (Italy)
Arie Croitoru, The University of Alberta (Canada)
Joep Crompvoets, Katholieke Universiteit Leuven (Belgium)
Leila De Floriani, University of Genova (Italy)
Michel Deshayes, Cemagref - UMR TETIS (France)
Juergen Döllner, HPI-Institute an der Universität Potsdam (Germany)
Matt Duckham, University of Melbourne (Australia)
Sara Fabrikant, University of Zürich (Switzerland)
Peter Fisher, University of Leicester (United Kingdom)
Anders Friis-Christensen, National Survey and Cadastre (Denmark)
Michael Gould, ESRI (USA)
Francis Harvey, University of Minnesota (USA)
Jan Haunert, University of Würzburg (Germany)
Gerard Heuvelink, Wageningen University (The Netherlands)
Stephen Hirtle, University of Pittsburg (USA)
Hartwig Hochmair, University of Florida (USA)
Bin Jiang, University of Gävle (Sweden)
Chris Jones, Cardiff University (United Kingdomr)
Didier Josselin, Université d'Avignon et des Pays du Vaucluse (France)
Maarit Kahila, Aalto University (Finland)
Derek Karssenberg, University of Utrecht (The Netherlands)
Marinos Kavouras, National Technical University of Athens (Greece)
Richard Kingston, University of Manchester (United Kingdom)
Eva Klien, Frauenhofer Institute for Computer Graphics - IGD (Germany)
Thomas Kolbe, Technical University Berlin (Germany)
Menno-Jan Kraak, Technical University Enschede – ITC (The Netherlands)
Marc van Kreveld, University of Utrecht (The Netherlands)
Antonio Krüger, University of Münster (Germany)
Lars Kulik, University of Melbourne (Australia)
Roger Longhorn, Geo:Connexion (United Kingdom)
Michael Lutz, Joint Research Centre (Italy)
Hans-Gerd Maas, Dresden University of Technology (Germany)
Stephan Maes, Universität der Bundeswehr München (Germany)
Bela Markus, University of West Hungary (Hungary)
Filipe Meneses, University of Minho (Portugal)

Adriano Moreira, University of Minho (Portugal)
Pedro Muro Medrano, Universidad de Zaragoza (Spain)
Javier Nogueras Iso, Universidad de Zaragoza (Spain)
Atsuyuki Okabe, University of Tokyo (Japan)
Peter van Oosterom, Technical University Delft (The Netherlands)
Volker Paelke, Leibniz Universität Hannover (Germany)
Marco Painho, New University of Lisbon (Portugal)
Dieter Pfoser, Research Academy Computer Technology Institute (Greece)
Lutz Pluemer, Universität Bonn (Germany)
Poulicos Prastacos, Foundation for Research and Technology, Heraklion (Greece)
Florian Probst, SAP Research CEC Darmstadt (Germany)
Hardy Pundt, University of Applied Sciences Harz (Germany)
Ross Purves, University of Zürich (Switzerland)
Martin Raubal, University of California at Santa Barbara (USA)
Tumasch Reichenbacher, University of Zürich (Switzerland)
Femke Reitsma, University of Canterbury (New Zealand)
Claus Rinner, Ryerson University (Canada)
Jorge Rocha, University of Minho (Portugal)
Maribel Yasmina Santos, University of Minho (Portugal)
Tapani Sarjakoski, Finnish Geodetic Institute (Finland)
Tiina Sarjakoski, Finnish Geodetic Institute (Finland)
Johannes Schöning, University of Muenster (Germany)
Monika Sester, Leibniz Universität Hannover (Germany)
Takeshi Shirabe, Technical University Vienna (Austria)
Elisabete Silva University of Cambridge (United Kingdom)
Spiros Skiadopoulos, University of Peloponnese (Greece)
Hans Skov-Petersen, University of Copenhagen (Denmark)
John Stell, University of Leeds (United Kingdom)
Kathleen Stewart Hornsby, The University of Iowa (USA)
Juan Suárez, Forestry Commission (United Kingdom)
Danny Vandenbroucke, Katholieke Universiteit Leuven (Belgium)
Agnès Voisard, Fraunhofer ISST (Germany)
Monica Wachowicz, University of New Brunswick (Canada)
Robert Weibel. University of Zürich (Zwitserland)
Stephan Winter, The University of Melbourne (Australia)
Mike Worboys, University of Maine (USA)
Bisheng Yang, Wuhan University (China)
May Yuan, University of Oklahoma (USA)
Francisco Javier Zarazaga-Soria, University of Zaragoza (Spain)

Contributing Authors

Nathalie Abadie
Université Paris-Est – Institut Géographique National (IGN) Saint-Mandé, France

Rafael Odon de Alencar
Database Laboratory, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil & SERPRO, Brazil

AliAsghar Alesheikh
Department of Geospatial Information Systems, K.N.Toosi University of Technology, Tehran, Iran.

Fadi Badra
Cemagref & UMR TETIS Montpellier, France

Felix Bache
Institute for Geoinformatics, University of Muenster, Germany & 52°North Initiative for Geospatial Open Source Software, Germany

Thomas Bartoschek
Institute for Geoinformatics, University of Muenster, Germany

Agnès Bégué
CIRAD & UMR TETIS Montpellier, France

Carmen Brando
Université Paris-Est – Institut Géographique National (IGN)-COGIT Laboratory, Saint-Mandé, France

Juliane Brink
Institute for Geoinformatics, University of Muenster, Germany

Arne Broering
Faculty ITC, University of Twente, The Netherlands & Institute for Geoinformatics, University of Muenster, Germany & 52°North Initiative for Geospatial Open Source Software, Germany

Bregje Brugman
Delft University of Technology (OTB – Department of GIS Technology), The Netherlands

Bénédicte Bucher
Université Paris-Est – Institut Géographique National (IGN)-COGIT Laboratory, Saint-Mandé, France

Françoise Chaillou
CERMA laboratory UMR CNRS Nantes, France

Padraig Corcoran
Department of Computer Science, National University of Ireland Maynooth. Ireland

Rodrigo Costa Mateus
Informatics Center, Federal University of Pernambuco, Recife, Brazil

Ron Dalumpines
TransLAB: Transportation Research Lab, School of Geography & Earth Sciences, McMaster University, Hamilton, Canada

Clodoveu Augusto Davis Jr.
Database Laboratory, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Demetris Demetriou
School of Geography, University of Leeds, United Kingdom

Laura Díaz
Institute of New Imaging Technologies, University Jaume I, Castellón, Spain

Patrice Dumas
Centre International de Recherche sur l'Environnement et le Développement (CIRED) Nogent-sur-Marne, France & CIRAD France

Cristina Dutra de Aguiar Ciferri
Computer Science Department, University of São Paulo at São Carlos, Brazil

Corné P.J.M. van Elzakker
Faculty ITC, University of Twente, The Netherlands

Manuel Fabritius
Fraunhofer IAIS, St. Augustin, Germany

Elisabetta Genovese
Centre International de Recherche sur l'Environnement et le Développement (CIRED) Nogent-sur-Marne, France

Carlos Granell
Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain

Elias Grinias
Department of Geoinformatics and Surveying, TEI of Serres, Greece

Stéphane Hallegatte
Centre International de Recherche sur l'Environnement et le Développement (CIRED) Nogent-sur-Marne, France & Ecole Nationale de la Météorologie, Meteo, France

Mahdi Hashemi
Department of Geospatial Information Systems, K.N. Toosi University of Technology, Tehran, Iran.

Jan-Henrik Haunert
Chair of Computer Science I, University of Würzburg, Germany

Dirk Hecker
Fraunhofer IAIS, Sankt Augustin, Germany

Joaquín Huerta
Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain

Christine Körner
Fraunhofer IAIS, Sankt Augustin, Germany

Dimitris Kotzinos
Department of Geoinformatics and Surveying, TEI of Serres, Greece & Institute of Computer Science, Foundation for Research and Technology – Hellas, Heraklion, Greece

Thomas Leduc
CERMA laboratory UMR CNRS Nantes, France -

Udo Lipeck
Institute of Practical Computer Science, Leibniz Universität Hannover, Germany

Ina Ludwig
Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS), Sankt Augustin, Germany

Thiago Luís Lopes Siqueira
São Paulo Federal Institute of Education, Science and Technology, IFSP, Brazil & Computer Science Department, Federal University of São Carlos, Brazil

Michael May
Fraunhofer IAIS, Sankt Augustin, Germany

Martijn Meijers
Delft University of Technology , OTB – Department of GIS Technology, The Netherlands

Henry Michels
Institute for Geoinformatics, University of Muenster, Germany

Michael Mock
Fraunhofer IAIS, St. Augustin, Germany

Peter Mooney
Department of Computer Science, National University of Ireland Maynooth. Ireland

Roland Müller
Fraunhofer IAIS, St. Augustin, Germany

Christoph Mülligann
Institute for Geoinformatics, University of Muenster, Germany

Tomoyuki Naito
Department of Mechanical and Environmental Informatics, Tokyo Institute of Technology, Japan

Daniel Nüst
Institute for Geoinformatics, University of Muenster, Germany

Peter van Oosterom
Delft University of Technology, OTB – Department of GIS Technology, The Netherlands

Jens Ortmann
Institute for Geoinformatics, University of Muenster, Germany

Toshihiro Osaragi
Department of Mechanical and Environmental Informatics, Tokyo Institute of Technology, Japan

Thomas Ouard
CERMA laboratory UMR CNRS Nantes, France

Edzer Pebesma
Institute for Geoinformatics, University of Muenster, Germany

Yoann Pitarch
LIRMM - CNRS - UM2, Montpellier, France

Frans Rip
Centre for Geo-Information, Wageningen University, The Netherlands

Ricardo Rodrigues Ciferri
Computer Science Department, Federal University of São Carlos, Brazil

Sven Schade
Institute for Environment and Sustainability, European Commission, Joint Research Centre, Ispra, Italy

Daniel Schulz
Fraunhofer IAIS, Sankt Augustin, Germany

Angela Schwering
Institute for Geoinformatics, University of Muenster, Germany

Darren M. Scott
TransLAB: Transportation Research Lab, School of Geography & Earth Sciences, McMaster University, Hamilton, Canada

Linda See
School of Geography, University of Leeds, United Kingdom & Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria

Monika Sester
Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany

Hendrik Stange
Fraunhofer IAIS, Sankt Augustin, Germany

Christoph Stasch
Institute for Geoinformatics, University of Muenster, Germany

John Stillwell
School of Geography, University of Leeds, United Kingdom

Alain Tamayo
Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain

Maguelonne Teisseire
Cemagref & UMR TETIS Montpellier, France

Frank Thiemann
Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany

Theo Tijssen
Delft University of Technology, OTB – Department of GIS Technology, The Netherlands

Valéria Cesário Times
Informatics Center, Federal University of Pernambuco, Recife, Brazil

Maike Krause-Traudes
Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS), Sankt Augustin, Germany

Pablo Viciano
Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain

Elodie Vintrou
CIRAD & UMR TETIS Montpellier, France

Angi Voss
Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS), Sankt Augustin, Germany

Jia Wang
Institute for Geoinformatics, University of Muenster, Germany

Hendrik Warneke
Institute of Practical Computer Science, Leibniz Universität Hannover, Germany

Table of Contents

GeoSensor Development and Application

Socio-Spatial Modeling and Analysis

Spatial Data Processing and Structuring

GI-Information Generation and Dissemination

Spatial-Temporal Modeling and Analysis

Spatio-Temporal Analysis of Tehran's Historical Earthquakes Trends

Mahdi Hashemi, AliAsghar Alesheikh

Department of Geospatial Information Systems, K.N. Toosi University of Technology, Tehran, Iran
m.hashemi1987@gmail.com, alesheikh@kntu.ac.ir

Abstract. Iran is one of the most seismically active regions of the globe. The metropolis of Tehran is located at the southern foothills of the Alborz Mountains that are the northern branch of the Alpine-Himalayan orogeny in Iran. The extremely high density of population concentrated in the Tehran metropolis (with more than 12 million inhabitants) coupled with the fragility of houses and life-lines, highlight the urgency of a reliable assessment of fault activity in this city. Three main active fault zones exist in the vicinity of Tehran: North Tehran fault; Mosha fault; and Eyvanekey-Kahrizak fault. In this paper, a total of 894 historical earthquakes of the study area with magnitudes over 2.5 Richter were collected since 1900. Three scenarios were considered for faults. In each scenario, every earthquake was associated to its relevant fault, spatial and temporal analyses were done to reveal the spatial and temporal trend of fault activities. First the north-south and east-west trends of magnitudes of earthquakes were verified and showed no meaningful trends. Spatial dependence of magnitudes of earthquakes was described in terms of global Moran's I and general G. The indices showed that the magnitudes of earthquakes were not clustered or spatially correlated. Ripley's K function determined that earthquakes are clustered at multiple distances. The temporal analyses were used to extract temporal trends in each scenario. The results showed that eastern sections of all faults are more active and the majority of large earthquakes have occurred in the middle sections of the faults. It is also anticipated that the eastern section of the Mosha fault is more capable of generating large earthquakes than the other faults in the Tehran region. The

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_1,

results of this paper can be useful for extracting hazardous areas and risk zonation or forecasting earthquakes.

1 Introduction

An earthquake is a rupture within the Earth caused by stress. Earthquakes occur principally by the sudden displacement on faults, when there is a build-up of stress in the crust caused by plate movement at a subduction zone or other fault line (Reiter 1991). Earthquakes have a greater effect on society than most people think. These effects range from economical to structural to mental. An earthquake only occurs for a few brief moments; the aftershocks can continue for weeks; the damage can continue for years.

The Iranian Plateau which is characterized by active faulting, active folding, recent volcanic activities, mountainous terrain, and variable crustal thickness, has been frequently struck by catastrophic earthquakes with high death tolls (Yaghmaei-Sabegh & Lam 2010). Seismicity here is the direct evidence of the continental convergence between the Arabian and the Eurasian plates (Martini et al. 1998; Doloei & Roberts 2003). In recent years, it has become obvious to all the professionals working towards reducing the losses due to earthquakes that a holistic approach in risk reduction is the only possible way of creating earthquake-resistant communities (Rose & Lim 2002).

The analysis of seismic activity variations with space and time is a complex problem (Jafari 2010). Geographical information analysis is the study of techniques and methods to enable the representation, description, measurement, comparison, and generation of spatial patterns (O'Sullivan & Unwin 2003).

Spatial data are related by their distances and spatial arrangements and characterized by spatial dependence and spatial heterogeneity (Ping et al. 2004). Tobler's (1979) first law of geography states: ''everything is related to everything else, but near things are more related than distant things'' (Ping et al. 2004). Spatial dependence is usually described by spatial autocorrelation using statistics such as Moran's I (Moran 1950), general G (Lloyd 2007), and Ripley's K function (O'Sullivan & Unwin 2003). Spatial heterogeneity relates to the spatial or regional differentiation, which follows the intrinsic uniqueness of each location (Anselin 1988). Cliff and Ord (1981) defined spatial autocorrelation as the phenomenon of systematic variability in a variable. Spatial autocorrelation exists when there is significant similarity/dissimilarity between the values of variables at all pairs of locations (Ping et al. 2004).

In this paper, geostatistical analyses are used to identify the activity of faults by location and time using available historical earthquakes in the area. As the first step, the historical earthquakes since 1900 were gathered and their relation to active faults in the area were verified. With 894 available earthquakes with magnitude over 2.5, it was found that the cumulative frequency of earthquakes based on the distance to the nearest active fault is a logarithmic graph. Considering the graph, all earthquakes - occurred 7 km away from each corresponding fault - were used to identify that fault's activity. For each fault scenario, its historical earthquakes were analyzed to study their spatial and temporal distribution. In addition, the earthquakes were studied to find, if they are clustered with/without considering their magnitudes. Figure 1. illustrates the general workflow of this study.

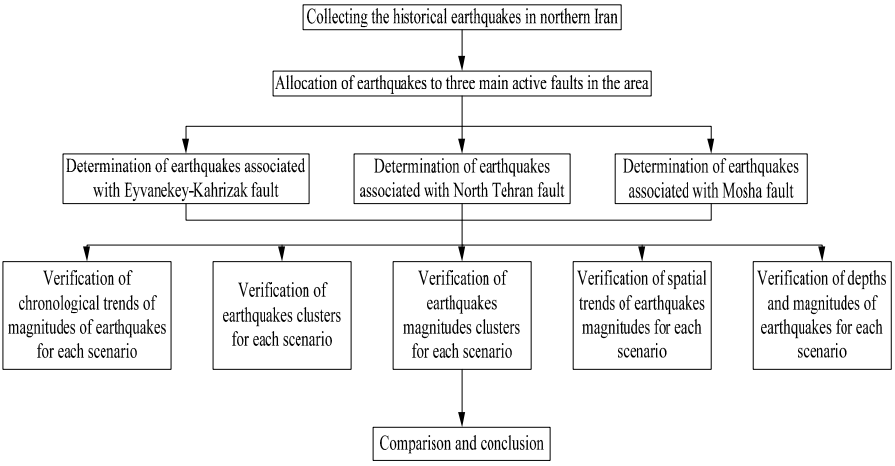


Fig. 1. General workflow of the study

1.1 Research background

Numerous researches have been made in analyzing earthquakes through geostatistics. Jafari (2010) made an attempt to fit six different statistical models (Exponential, Gamma, Lognormal, Pareto, Rayleigh, Weibull) to the time series generated by the earthquakes after 743 AD around Tehran to estimate recurrence times and conditional probability of the next great earthquake. He predicted that a large damaging earthquake may occur around Tehran approximately every 10 years. Zafarani and his colleagues (2009) applied stochastic finite-fault modeling to simulate horizontal acceleration time histories for the rock and soil recordings of eight events that occurred in east-central and northern Iran during 1979-2005. The calibrated model was then used to predict ground motions in the Tehran met-

ropolitan area for future large earthquakes along three faults in the Alborz seismic zone. Hamzehloo and his partners (2007) estimated the expected ground motion in a very high seismicity zone of Tehran. They calculated that the maximum credible earthquake for faults around Tehran varies between 6.2 and 7.3 on the Richter scale through the length of the faults considered. Shabestari and his colleagues (2004) converted the ground motion values to those at a hypothetical ground base-rock level. Then, a kriging method, assuming an attenuation relationship at the base-rock as a Tehran component, was applied and finally, the spatial distribution of ground motion at ground surface was obtained by applying GIS-based amplification factors for the entire region. Khazai and Sitar (2003) used GIS to conduct a spatial characterization of the slope failures, including distribution of type, size, slope angle, bedrock geology, ground motion, and distance from earthquake source. Analysis of the Chi-Chi earthquake data suggested that only the very large catastrophic dip slope failures can be specifically tied to a particular geologic setting. All other types of landslides were most closely tied to the strong motion. Kamp and his team (2008) developed and analyzed a spatial database, which included 2252 landslides using ASTER satellite imagery and GIS technology. A multi-criteria evaluation was applied to determine the significance of event-controlling parameters in triggering the landslides. The parameters included lithology, faults, slope gradient, slope aspect, elevation, land cover, rivers, and roads. Jinhui and his colleagues (2010) presented a quantitative analysis of the number and area of the landslides triggered by the Wenchuan M_s 8.0 earthquake from Anxian to Beichuan. In their study, the controlling factors of landslides were: reliefs, slopes, and topographic roughness.

2 Materials and Methods

2.1 Study area

Tehran, the capital of Iran, is located at the foot slope of the Alborz Mountains forming parts of the Alpine-Himalayan orogenic zone and is limited to the north by the Alborz Mountains, to the south by Bibi Shahrbano Mountain, and to the east by Sepah Mountain (Yaghmaei-Sabegh & Lam 2010).

The occurrence of more than 10 earthquakes with magnitude around 5 during the 20th century in the Tehran vicinity indicates the intense activity of faults. Seismologists believe that a strong earthquake may strike Tehran

in the near future, given that the city has not experienced a major earthquake disaster since 1830 (Zafarani et al. 2009). Among the many active faults in the area, the most probable hazardous faults are the Mosha, Eyvanekey-Kahrizak, and North Tehran faults (Sadidkhouy et al. 2008).

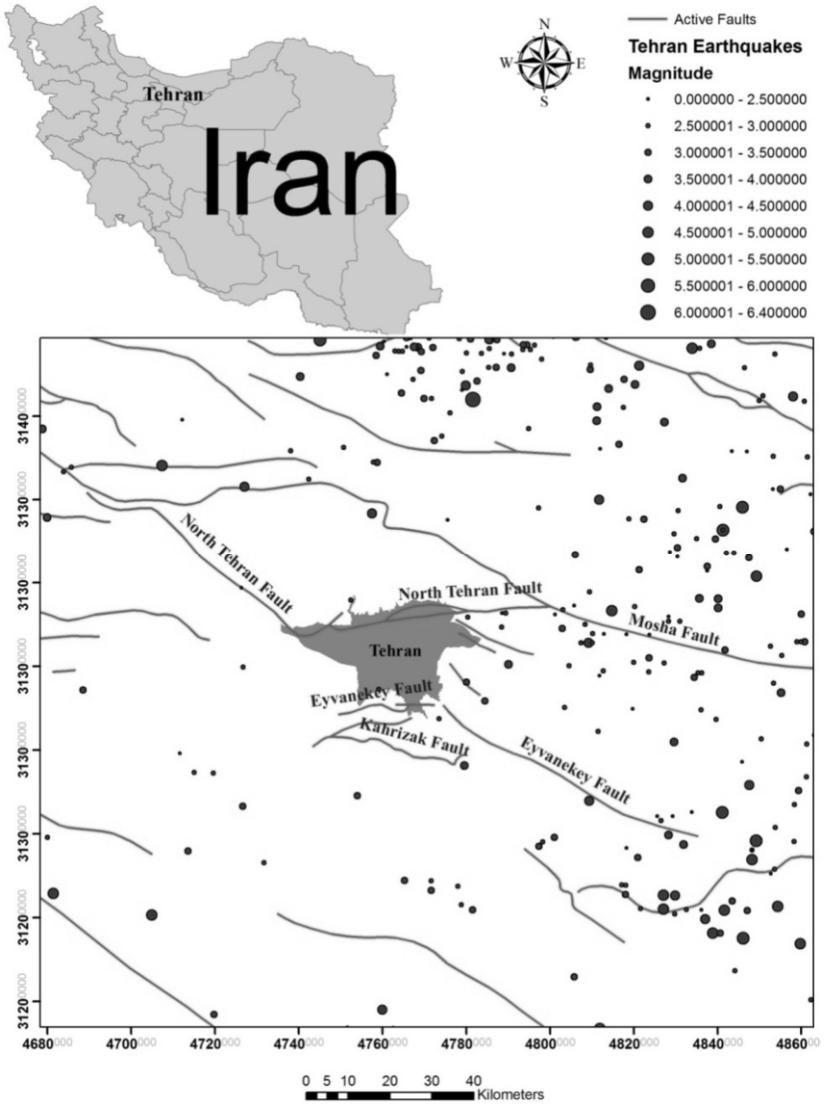


Fig. 2. Tehran city, active faults around it and earthquakes since 1900

The historical earthquakes of Tehran since 1900 and Iran's faults map were obtained from the International Institute of Earthquake Engineering

and Seismology (Iran) and were projected to Lambert Conformal Conic Projection. The study area is shown in Figure 2.

The Mosha fault is a major fault over 200 km long, and it consists of several segments (Final report of the study on seismic microzoning of the Greater Tehran 2000). The Mosha fault seems to be one of the most active, experiencing several earthquakes of magnitude greater than 6.5 in years 958, 1665, and 1830 (Ashtari et al. 2005). The earthquake in 1830 corresponded to activity on the eastern segment of the Mosha fault. The largest historical earthquake occurred in 958 with M_w=7.7, at about 50 km from the center of Tehran. This earthquake corresponded to activity on the western segment of the Mosha fault.

The North Tehran fault is located between the western segment of the Mosha fault and the city of Tehran. If the North Tehran fault is activated, the damage, which the resulting earthquake could cause, will be heavier than that which would be due to the re-occurrence of the event in 958. This fault extends over 90 km, but the northwestern part of it is far from the city of Tehran.

The Kahrizak fault is located south of the city of Tehran, and its length is approximately 20 km. The Kahrizak (south Ray) fault extends along the south side of the Ray depression. The interval between these two faults is only 3 to 5 km. It is considered that the root of these two faults is the same and they are branches of one fault.

2.2 Assigning earthquakes to each fault

A total of 894 historical earthquakes with magnitudes over 2.5 in northern Iran – the vicinity of Tehran – were collected. The cumulative frequency of earthquakes based on the distance to the nearest active fault is shown in Figure 3.

The graph in Figure 3 is logarithmic and shows that the earthquakes have occurred near the faults. So, the earthquakes can be classified based on faults. Every earthquake is assigned to its nearest active fault, but to ensure that earthquakes are not attributed to faults mistakenly, only half of the total earthquakes that are closer to the three faults are associated to them. So a threshold of 7 kilometers has been observed in assigning every earthquake to its closest fault. This ensures that the majority of the earthquakes are associated to its close-by fault. The earthquakes were then used for geostatistical tests to verify the activities of that specific fault

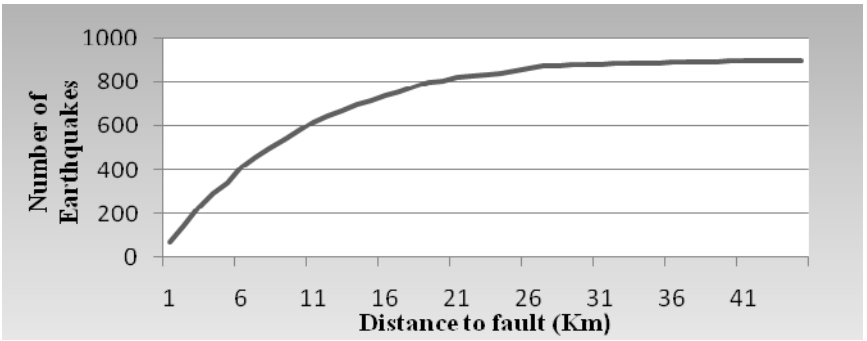


Fig. 3. The cumulative frequency of earthquakes-distance to nearest active fault graph

3 Discussion on Faults Scenarios

Three scenarios are observed and tested in this study.

3.1 Mosha fault scenario

A total of 42 earthquakes in 40 years were associated to this fault. The depths of these earthquakes are between 6 and 78 km with a mean value of 17 km. They all are classified as shallow. The histogram of frequency of earthquakes based on their magnitudes is shown in Figure 4 The heights of the bars show the abundance of earthquakes for each class. Note that all magnitudes are less than 4.6 and the number of large earthquakes is less than weak earthquakes.

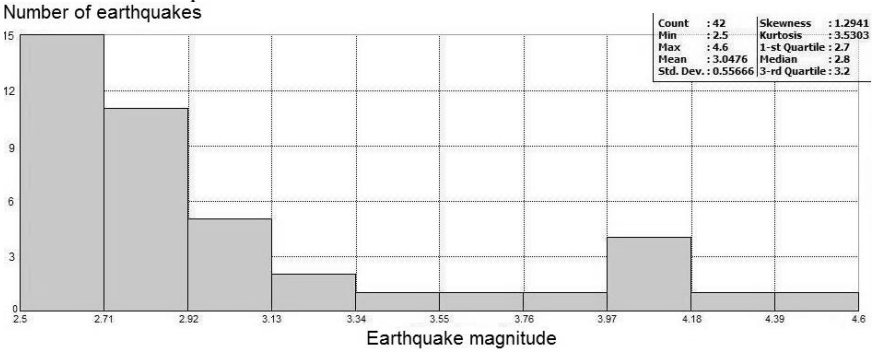


Fig. 4. The histogram of earthquakes magnitudes for Mosha fault scenario

In order to study the spatial trend of the magnitudes of the earthquakes associated to the Mosha fault, they were projected to a 3 dimensional space: two pivots show the locations of earthquakes and the 3rd dimension shows their magnitude. Then all the bars in Figure 5 are projected to both planes XZ and YZ. Finally, a curve is fitted to each plane's points. Each curve illustrates the magnitude trend along a pivot (Reimann et al. 2008). Figure 5 shows the spatial trend of earthquake magnitudes.

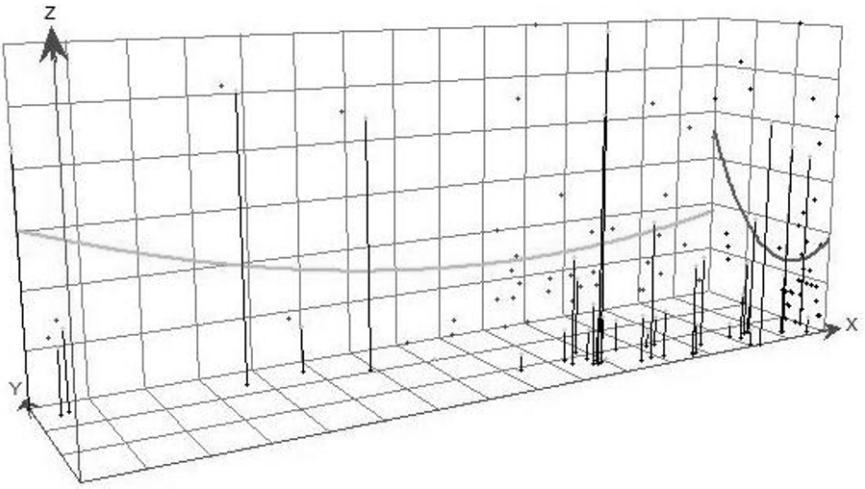


Fig. 5. The spatial trend of earthquake magnitudes for the Mosha fault scenario

Considering that the above fault is extended along northwest to southeast, Figure 5 demonstrates that the magnitudes decrease towards the southeast gradually and increase again at the end. The largest earthquakes are located in the middle section of the fault, whereas most earthquakes have occurred in the eastern section.

The verification of variogram of the magnitudes can reveal whether they are correlated (Reimann et al. 2008). Vertical axis in variogram shows semivariance for different distance lags. Semivarince for distance h – $\gamma(h)$ – is calculated from Eq. 3.1:

$$\gamma(h) = \frac{1}{2n}\sum_{i=1}^{n}\left\{z_{(x_i)} - z_{(x_{i+h})}\right\}^2 \qquad (3.1)$$

In the above Eq., n is the total number of earthquake pairs that are located within h, $z_{(x)}$ is the magnitude of earthquake x, and h is the distance between two earthquakes. Figure 6 shows the variogram of the magnitudes.

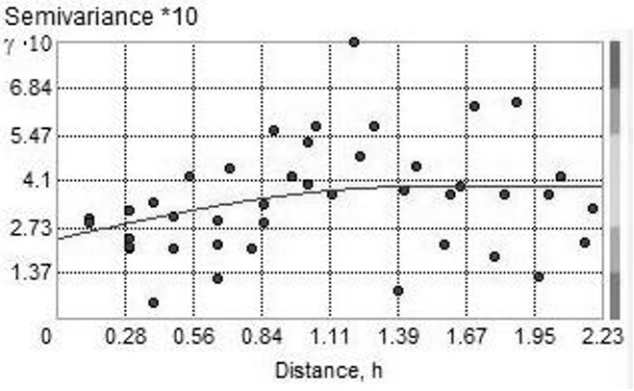


Fig. 6. The variogram of earthquake magnitudes for the Mosha fault scenario

The range of variogram is 1.4 km, the sill of variogram is 0.4, and the nugget is 0.23, which means the noise in magnitudes is high. Since the values of nugget and sill are very close, this variogram implies that the magnitudes of earthquakes do not have a meaningful spatial correlation.

To certify that the magnitudes of earthquakes are not clustered, the general G index and Moran's I index were used (Lloyd, 2007). The general G index is calculated from Eq. 3.2:

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j} z_i z_j}{\sum_{i=1}^{n}\sum_{j=1}^{n} z_i z_j}, \forall\, j \neq i \tag{3.2}$$

In the above Eq., z_i and z_j are the magnitudes of two earthquakes i and j, $w_{i,j}$ is the square of inverse distance between two earthquakes, and Z_G-score is computed from Eq. 3.3:

$$Z_G = \frac{G - E[G]}{\sqrt{V[G]}} \tag{3.3}$$

$$E[G] = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}}{n(n-1)}, \forall\, j \neq i \tag{3.4}$$

$$V[G] = E[G^2] - E[G]^2 \tag{3.5}$$

The general G index was estimated as 0.24 and the Z_G-score was equal to -0.18. These numbers confirmed the results of variogram that showed the distribution of earthquake magnitudes is statistically independent.

The Global Moran's I index was calculated from Eq. 3.6:

$$I = \frac{n}{S_0} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j} z_i z_j}{\sum_{i=1}^{n} z_i^2} \tag{3.6}$$

$$S_0 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j} \tag{3.7}$$

The Moran's I index was equal to -0.04 which means the magnitudes are not clustered; they are more randomly distributed. Therefore, previous results are asserted. This concept is shown in Figure 7.

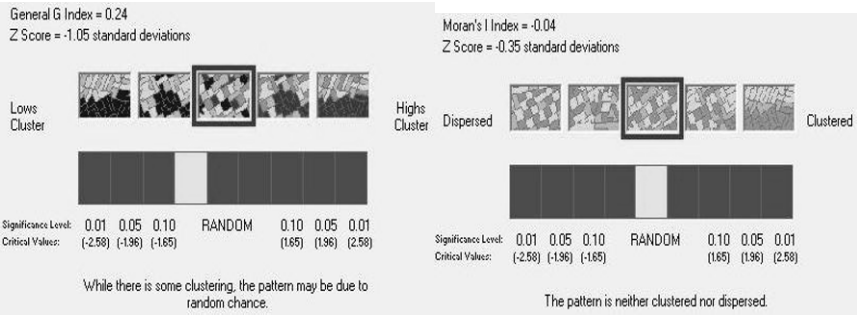


Fig. 7. The general G index (left) and Moran's I index (right) for earthquake magnitudes in the Mosha fault scenario

To verify if the earthquakes are clustered, the Ripley's K function was used (O'Sullivan & Unwin 2003). The merit of this index is considering the distance between an earthquake and all other earthquakes. The amount of this index for distance d is calculated from Eq. 3.8:

$$k(d) = \sqrt{\frac{A\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} w(i,j)}{\pi n(n-1)}} \tag{3.8}$$

In the above Eq., n is the total number of earthquakes and A is the area of the study region. If the distance between two earthquakes i and j is less than d, then w will be one, otherwise it is zero. This function illustrated that earthquakes are extremely clustered (Figure 8). Note that previous results showed that the magnitudes of earthquakes are not spatially correlated but Ripley's K function showed that the earthquakes are clustered and these results are not in contrast. There is a large cluster of earthquakes at the eastern section of the Mosha fault.

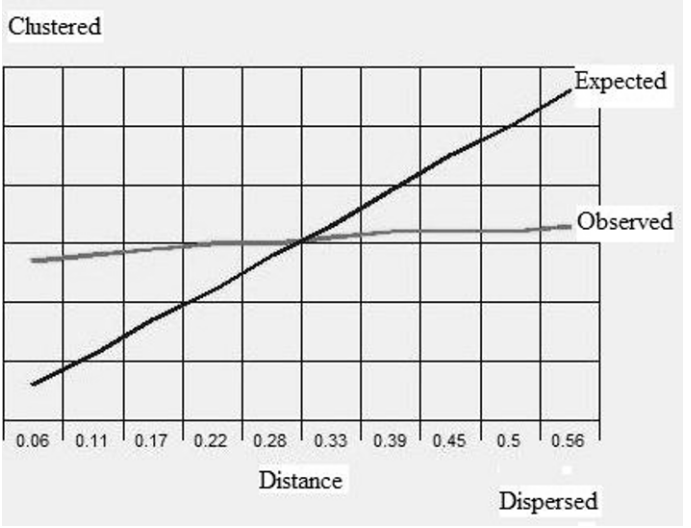


Fig. 8. The Ripley's K function for earthquakes associated to the Mosha fault

The magnitudes based on the occurrence date of earthquakes graph can show the chronological activities of the fault. Such a graph for the Mosha fault is shown in Figure 9.

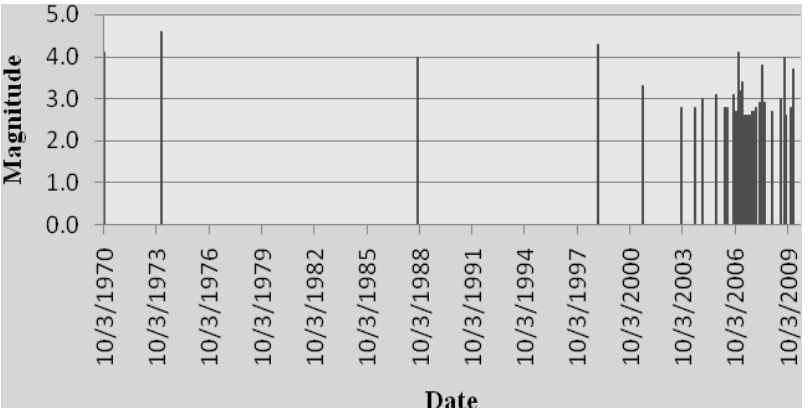


Fig. 9. The magnitudes based on the occurrence date of Mosha fault earthquakes

There is a gap between 1973 and 1988 and another one between 1988 and 1998; the movements were not recorded during these periods. So this graph illustrates that the Mosha fault has had a uniform and continuous

performance and has not created any large earthquakes during the last 40 years.

3.2 North Tehran fault scenario

A total of 9 earthquakes with depths between 6 to 18 km and a mean of 13.5 km are associated to this fault during the last 10 years. Because of their shallow depths, the earthquakes of this fault can cause vast destructions in a small area (Coburn & Spence 2002), but fortunately the magnitudes are less than 3.2. The histogram of magnitudes is shown in Figure 10.

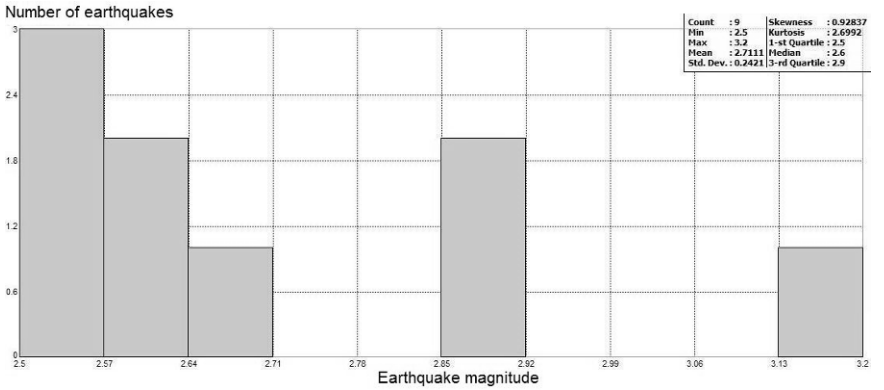


Fig. 10. The histogram of earthquake magnitudes for the North Tehran fault scenario

The magnitude trend graph of the earthquakes associated to this fault (Figure 11) illustrates an increase in magnitudes towards the east and south. The east section of this fault is more active and has created the largest earthquakes.

Due to the low number of earthquakes, the verification of variogram is not effective (Reimann et al. 2008). The general G index and global Moran's I index for earthquake magnitudes are 0.61 and -0.2 respectively, indicating that the magnitudes are not clustered about this fault too. These two indices are shown in Figure 12.

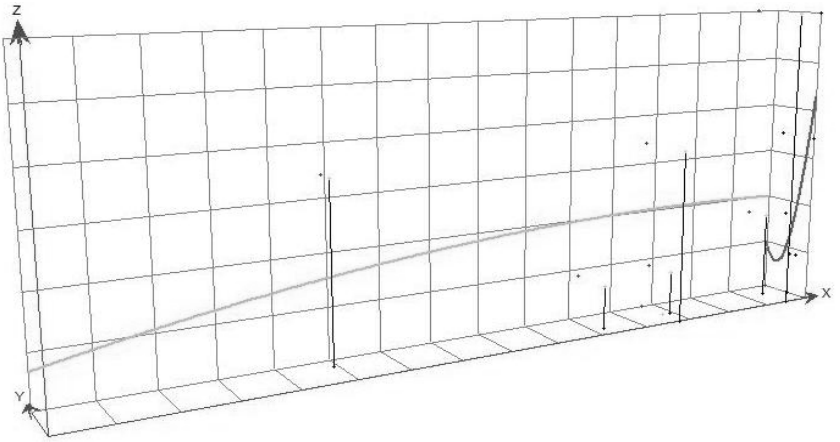


Fig. 11. The spatial trend of earthquake magnitudes for the North Tehran fault scenario

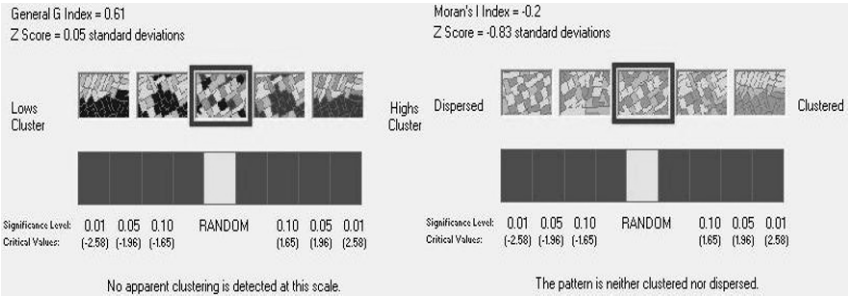


Fig. 12. The general G index (left) and Moran's I index (right) for earthquake magnitudes in the North Tehran fault scenario

The Ripley's K function showed that there are some clusters of earthquakes at the eastern section. The chronological trend of the magnitudes of earthquakes (Figure 13) represents the more activities of this fault during 2006 to 2008. Figure 13 shows the activation of the North Tehran fault in two-year intervals. It is, then, anticipated that the activity of the fault that has started in 2010, continues until 2012. Of course, the magnitudes of the earthquakes associated to this fault cannot cause severe destructions.

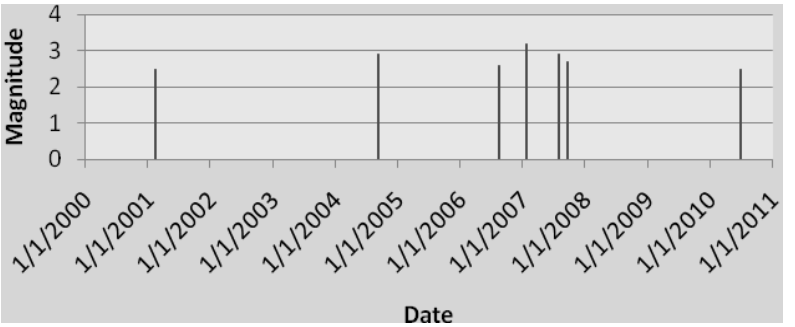


Fig. 13. The magnitudes based on the occurrence date of the North Tehran fault earthquakes

3.3 Eyvanekey-Kahrizak fault scenario

This fault has not had much activity during the 50 past years. A total of 10 very shallow earthquakes were associated to this fault. Their depths are between 11 and 16 km. There is only one earthquake with a depth of 144 km and a magnitude of 4.5 in 1967. The magnitudes in the 50 past years are less than 4.5 and they could not cause destructions. The histogram of magnitudes is presented in Figure 14.

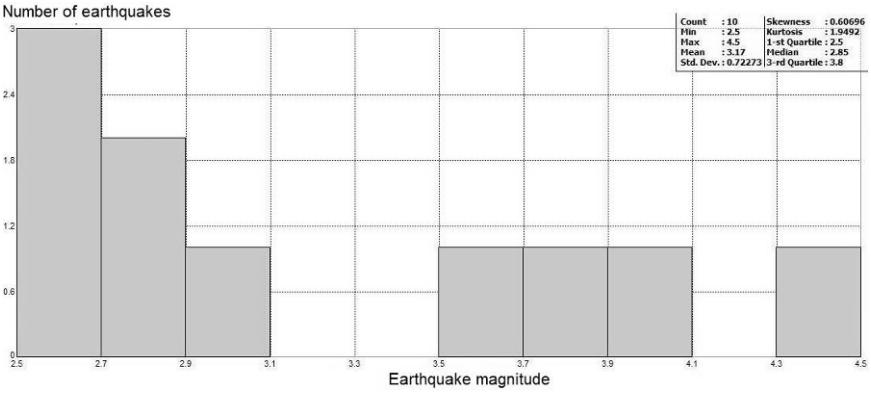


Fig. 14. The histogram of earthquake magnitudes for the Eyvanekey-Kahrizak fault scenario

The spatial trend of magnitudes is shown in Figure 15. The verification of magnitude trend of earthquakes indicates that the middle section of the

Eyvanekey-Kahrizak fault has created larger earthquakes and can be destructive.

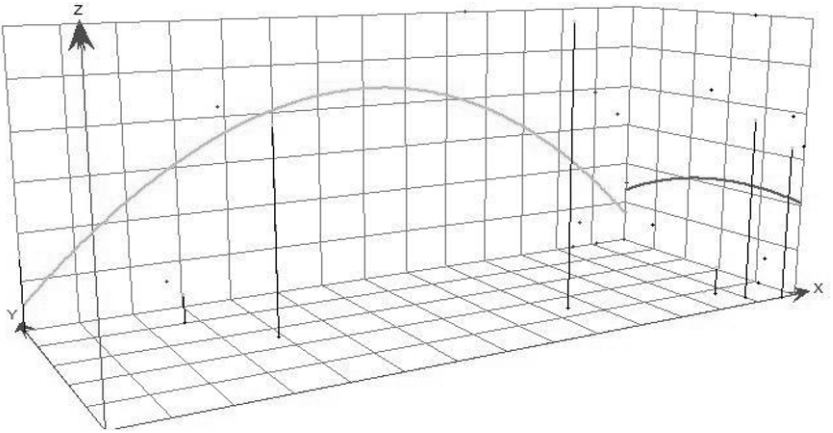


Fig. 15. The spatial trend of earthquake magnitudes for the Eyvanekey-Kahrizak fault scenario

The general G index is 0.35 and global Moran's I index is -0.14 which means there is no correlation between magnitudes of this earthquake scenario like its predecessors (Lloyd 2007). However, the earthquakes are extremely clustered in the eastern section. These indices are shown in Figure 16.

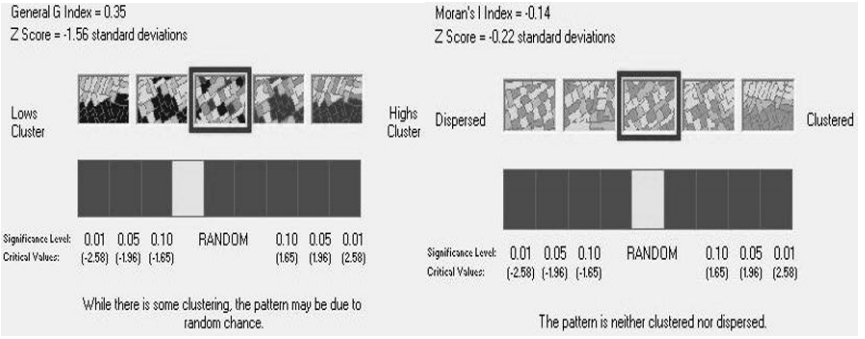


Fig. 16. The general G index (left) and Moran's I index (right) for earthquake magnitudes in the Eyvanekey-Kahrizak fault scenario

In order to study the chronological trend of magnitudes, the earthquake with magnitude 4.5 occurred in 1967 was excluded from further analysis, because no data is available after that until year 2000 (Figure 17). This graph indicates an increase in activity of the fault after 2000. But the

depths and magnitudes of earthquakes are very low. The proximity of this fault to the southern parts of Tehran, where buildings are very vulnerable and soil is soft (Shafiee & Azadi 2007), increases the sensitivity to the issue.

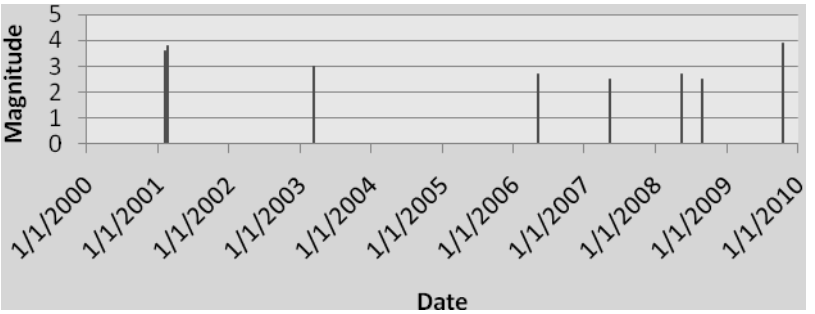


Fig. 17. The magnitudes based on the occurrence date of the Eyvanekey-Kahrizak fault earthquakes

4 Conclusions

In this paper, three main faults around the highly populated metropolis of Tehran (Mosha, North Tehran, Eyvanekey-Kahrizak faults) were considered as sources of earthquakes in the area. All the faults have shallow historical earthquakes. Geostatistical tests demonstrated that the magnitudes of earthquakes are not correlated but the earthquakes themselves are extremely correlated and clustered considering their location. The results of this paper confirm that if very light earthquakes occur in the study area, large earthquakes will be followed afterwards. Note that the hazard zone is not a point, but it is a region with many small and large historical earthquakes because of the spatial correlation between earthquakes.

The spatio-temporal analyses indicated that the eastern sections of all faults are more active than the other parts. The middle section of the Mosha and Eyvanekey-Kahrizak faults and the eastern and southern sections of the North Tehran fault have created the largest earthquakes.

The North Tehran fault has not had much activity during the last 10 years. It, however, created weak earthquakes in two-year intervals. The activity of Eyvanekey-Kahrizak fault has increased during recent years. Considering the low depths of its earthquakes and its proximity to southern Tehran – where buildings are very vulnerable and soil is soft – increases

the threat of this fault, but fortunately the magnitudes of its earthquakes have been less than 4.5 so far.

The Mosha fault is much more active than the other faults in the area and also very far from the city (about 50 km from the center of Tehran). This fault has had many uniform activities during the last 40 years and has continuously created very weak earthquakes. The spatio-temporal analyses of the mentioned fault activities proved that the eastern section of the Mosha fault is more active. So, considering a weak correlation among magnitudes of earthquakes and a powerful correlation between earthquake occurrences, it is concluded from this paper that the eastern section of the Mosha fault is more capable of generating large earthquakes than the other faults in the Tehran region.

References

Anselin L (1988) Spatial econometrics: methods and models. Kluwer Academic Publishers

Ashtari M, Hatzfeld D, Kamalian N (2005) Microseismicity in the region of Tehran. J Tectonophysics 395:193– 208

Cliff A, Ord J (1981) Spatial processes. Pion

Coburn A, Spence R (2002) Earthquake Protection (2nd edn). Wiley

Doloei J, Roberts R (2003) Crust and uppermost mantle structure of Tehran region from analysis of teleseismic P-waveform receiver functions. J Tectonophysics 364: 115–133

Japan International Cooperation Agency (JICA) & Center for Earthquake and Environmental Studies of Tehran (CEST). (2000) Final report of the study on seismic microzoning of the Greater Tehran

Hamzehloo H, Vaccari F, Panza G (2007) Towards a reliable seismic microzonation in Tehran, Iran. J Engineering Geology 93: 1–16

Jafari M A (2010) Statistical prediction of the next great earthquake around Tehran, Iran. J Geodynamics 49: 14–18

Jinhui Y, Jie C, XiWei X, Xulong W, Yonggang Z (2010) The characteristics of the landslides triggered by the Wenchuan Ms 8.0 earthquake from Anxian to Beichuan. J Asian Earth Sciences 37: 452–459

Kamp U, Growley BJ, Khattak GA, Owen LA (2008) GIS-based landslide susceptibility mapping for the 2005 Kashmir earthquake region. J Geomorphology 101: 631–642

Khazai B, Sitar N (2003) Evaluation of factors controlling earthquake-induced landslides caused by Chi-Chi earthquake and comparison with the Northridge and Loma Prieta events. J Engineering Geology 71:79–95

Lloyd CD (2007) Local models for spatial analysis.CRC Press, Boca Raton

Martini PD, Hessami K, Pantosti D, D'Addezio G, Alinaghi H, Ghafory-Ashtiani M (1998) A geologic contribution to the evaluation of the seismic potential of the Kahrizak fault (Tehran, Iran). J Tectonophysics 287: 187- I99

Moran P (1950) Notes on continuous stochastic phenomena. J Biometrika 37: 17–23

O'Sullivan D, Unwin D (2003) Geographic information analysis. Wiley, New Jersey

Ping J, Green C, Zartman R, Bronson K (2004) Exploring spatial dependence of cotton yield using global and local autocorrelation statistics. J Field Crops Research 89: 219–236

Reimann C, Filzmoser P, Garrett RG, Dutter R (2008) Statistical data analysis explained: Applied environmental statistics with R. Wiley

Reiter L (1991) Earthquake hazard analysis: issues and insights. Columbia University Press

Rose A, Lim D (2002) Business interruption losses from natural hazards: conceptual and methodological issues in the case of the Northridge earthquake. J Environmental Hazards 4: 1–14

Sadidkhouy A, Javan-Doloei G, Siahkoohi H (2008) Seismic anisotropy in the crust and upper mantle of the Central Alborz Region, Iran. J Tectonophysics 456: 194–205

Shabestari K T, Yamazaki F, Saita J, Matsuoka M (2004) Estimation of the spatial distribution of ground motion parameters for two recent earthquakes in Japan. J Tectonophysics 390: 193– 204

Shafiee A, Azadi A (2007) Shear-wave velocity characteristics of geological units throughout Tehran City, Iran. J Asian Earth Sciences 29: 105–115

Yaghmaei-Sabegh S, Lam NT (2010) Ground motion modelling in Tehran based on the stochastic method. J Soil Dynamics and Earthquake Engineering 30: 525–535

Zafarani H, Noorzad A, Ansari A, Bargi K (2009) Stochastic modeling of Iranian earthquakes and estimation of ground motion for future earthquakes in Greater Tehran. J Soil Dynamics and Earthquake Engineering 29: 722–741

Damage Assessment from Storm Surge to Coastal Cities: Lessons from the Miami Area

Elisabetta Genovese[1], Stéphane Hallegatte[1,2], Patrice Dumas[1,3]

[1]Centre International de Recherche sur l'Environnement et le Développement (CIRED)
[2]Ecole Nationale de la Météorologie, Meteo France
[3]Centre de Coopération Internationale en Recherche Agronomique pour le Développement (CIRAD)
genovese@centre-cired.fr, hallegatte@centre-cired.fr, dumas@centre-cired.fr

Abstract. Coastal cities are growing at a very rapid pace, both in population and in terms of assets; therefore, flood risk is likely to increase substantially in these areas in the absence of specific protections. In addition, great uncertainty surrounds the future evolution of hurricane intensity and sea level rise. The area of Miami represents a clear hotspot of human and economic coastal flood exposure: there are more than 5 million inhabitants in the Miami metropolitan area and the population is growing. It is also a low-lying city with most of the population living below an elevation of 10m and is located in a region where tropical cyclones hit frequently. The present study is focused on the two contiguous counties of Miami, Dade and Broward. In this analysis, we consider the impact of different storm surges predicted by the computerized model SLOSH[1] and investigate flood risks with current sea level, considering different hurricane parameters (storm category and direction, wind speed, and tide level). For each impact, we apply a damage function and determine if the considered storm surges potentially lead to asset loss, considering both properties and their contents. The results show that, in absence of protections, losses will be very high for large storm surges reaching up to tens of billions USD. In the second part of the analysis, we demonstrate how the economic impact

[1] http://www.nhc.noaa.gov/HAW2/english/surge/slosh.shtml

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_2,

changes when protections are built up, considering different dams' heights. We conclude that raising flood defences would be beneficial, since the consequences of a storm surge could be enormous.

1 Introduction

It is very likely that flood risks will increase in coastal cities in the next years, because of demographic, socio-economic, and environmental trends (Webster et al. 2005; Nicholls et al. 2007). The assessment of this increase is necessary in order to include the range of possible changes within urban and land-use planning (Lugeri et al. 2010). Moreover, urbanization and population in these areas are still growing at a very rapid pace, driven by economic opportunities and the development of international trade. Therefore, the product of an interaction between numerous aspects, such as climatic, socio-economic, and institutional, is increasing the risk of big damage losses (Lugeri et al. 2006) and it is suitable to reduce future risks through targeted territorial development plans. This article proposes the case of the Miami area illustrating a methodology to assess coastal flood risks in urban areas and it aimes to derive more general lessons, useful for all coastal cities.

Since 1990, Florida has been struck directly by 22 tropical storms and hurricanes. In 1992, Hurricane Andrew hit Dade County with Category 5 force, generating 17-foot (more than 5 meters) storm surges. 23 people were killed and property damage in the whole state of Florida from Andrew was estimated at 25.5 billion USD. The marine ecosystem, including the natural reef, was also heavily damaged. Between August and September 2004 several hurricanes struck the Florida coast (see Table 1). Eventually, 45 people were killed and estimated damages across the southeastern United States totalled over 21.1 billion USD[2].

Even before the recent hurricane seasons, 40% of Florida's beaches were listed as critically eroded. In 1986, the Florida Legislature adopted a complete beach management planning program to protect and restore the state's beaches[3]. Between 1994 and 2004, Florida began the largest and most costly beach and dune rebuilding program in US history: 242 million USD were spent on beach nourishment, aiming to absorb the wave energy dissipated across the surf zone. Following the catastrophes of 2004, there was a hurry to immediately restore damaged beaches. In 2004 and 2005,

[2] http://www.edf.org/article.cfm?contentid=5361
[3] http://www.dep.state.fl.us/beaches/programs/bcherosn.htm

the state spent approximately 173 million USD on sand[4] and over 582 million USD in 2006 for beach erosion control activities and hurricane recovery[5].

Table 1. Florida major hurricanes in the last 100 years[6].

Storm	Category	Year	Landfall Intensity	Landfall Location
Andrew	5	1992	145	Homestead
Labor Day	5	1935	160	Craig Key
Charley	4	2004	130	Cayo Costa
Donna	4	1960	120	Key Vaca
Unnamed	4	1949	130	Palm Beach Shores
Unnamed	4	1947	135	Pompano Beach, Florida
Unnamed	4	1945	120	Upper Florida Keys
Okeechobee	4	1928	130	Jupiter Island
Great Miami	4	1926	115	South Miami
Unnamed	4	1919	130	Offshore Florida Keys
Dennis	3	2005	105	Santa Rosa Island
Wilma	3	2005	105	Cape Romano
Ivan	3	2004	105	Gulf Shores, Alabama
Jeanne	3	2004	105	Hutchinson Island
Opal	3	1995	100	Pensacola Beach
Elena	3	1985	100	Gulfport, Mississippi
Eloise	3	1975	110	Bay County
Betsy	3	1965	110	Upper Florida Keys
Easy	3	1950	105	Cedar Key
King	3	1950	105	Miami
Unnamed	3	1948	105	Lower Florida Keys
Unnamed	3	1948	110	Marathon
Unnamed	3	1941	105	Goulds
Unnamed	3	1933	110	Jupiter
Unnamed	3	1917	100	Okaloosa Count

Despite the large amount of money invested, our study suggests that, in the case of storms with elevated water levels and high waves, beach nourishment does not provide adequate benefits in the form of storm damage

[4] http://www.surfrider.org/stateofthebeach/05-sr/state.asp?zone=se&state=fl&cat=bf

[5] http://www.dep.state.fl.us/beaches/programs/bcherosn.htm

[6] Atlantic hurricane research division (2008). "All U.S. Hurricanes (1851-2007)". NOAA. http://www.aoml.noaa.gov/hrd/hurdat/ushurrlist18512007.txt.

reduction and cannot be sufficient to avoid the water impact on structures and infrastructures.

The present study is focused on the area of Miami, which clearly represents a hotspot of human and economic coastal flood exposure (Herweijer et al. 2008). Its metropolitan area has a population of more than 5 million inhabitants. The number of inhabitants has grown by 35% since 1990 and it keeps growing; new residential and commercial constructions have been widespread.

According to an OECD global analysis of vulnerable coastal cities (Nicholls 2007), Miami is one of the port cities with the highest exposure and vulnerability to climate extremes in the world, even in the present situation. It is located in a region where tropical hurricanes hit frequently (see a statistical analysis of hurricane landfalls, in Hallegatte et al. 2007) and, in the future, it may be one of the most exposed areas to coastal flooding in terms of infrastructure and other assets.

Since Miami is also a low-lying city, with most of the population living below an elevation of 10 meters, hurricanes often cause significant storm surges and losses from these storms could be enormous in such a flat area. When considering its high exposure, the city has a surprisingly very low level of protection with no comprehensive seawall or dam system to protect the city from storm surges.

This paper focuses on current flood risks and describes the impacts of water-related risks in this region, specifically in the Miami Dade and Broward counties, with the aim to establish an overall cost-estimate of potential losses. In particular, the work focuses on the economic aspects of flood damages by investigating the value of physical assets affected by the event. To evaluate the cost of damages on direct losses in residential areas, we propose a damage assessment.

In the first part of this study, we analyse storm surge losses considering different hurricanes' intensities and directions, in order to estimate storm surge heights and winds, according to the result of the computerized model SLOSH[7]. Then, we assess the direct losses that could be caused by episodes of sea level rise at different levels, according to the economic values of insured properties provided by Risk Management Solution (RMS)[8]. This analysis is used to determine the benefits from protection, in the current situation, as a function of different storm surges. Finally, we determine the consequences of an adaptation strategy starting from the current condition and then analyse how the loss prospective can change when protections are added. The study demonstrates that storm surges will

[7] http://www.nhc.noaa.gov/HAW2/english/surge/slosh.shtml
[8] http://www.rms.com/

lead to definitive losses of assets and our conclusion is that to take action and to raise flood defences are urgently required. In a follow-up analysis, climate change and sea level rise will be included, to investigate how these additional drivers modify the optimal defence strategy.

2 The effects of climate change on sea levels and hurricanes

The 2007 Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC) projected that global mean sea levels would rise by 18 – 59 cm above 1990 levels by the 2090s. However, these projections do not fully include contributions from the melting ice sheets (due to the limitations of the modelling techniques used). Rahmstorf (2007) employs a technique that observes the relationship between global sea levels and temperature to project future sea levels from temperature projections. While very simplistic, this technique has the advantage of using real data and avoiding many of the uncertainties introduced through using global climate models. Rahmstorf (2007) projects that global sea levels could increase by around 50 – 140 cm above 1990 levels by 2100. Pfeffer et al. (2008) conclude that sea level rise cannot exceed 2 m in 2100, with a best guess at 80 cm.

Depending on the methodology and the model, hurricanes are predicted to become more intense, stable, or less frequent (see, e.g., Landsea 2005; Emanuel 2008). On top of climate-change-related changes in sea level, water height will continue to vary over time as a result of weather-related effects, including storm surges. Storm surge is water that is pushed toward the shore by the force of winds that swirl around the storm. This progressing surge combines with the normal tides to create the hurricane storm tide, which can increase the mean water level by 15 feet or more. Storm surge begins to grow when the hurricane is still far out at sea over deep water[9]. The low pressure near the centre of the storm causes the water to rise.

Climate change can also affect the amplitude of these variations by changing the frequency of the variability through, for example, changes in hurricane intensity. However, future modifications in water levels and hurricane intensities are still heatedly debated in the scientific community and cannot be easily anticipated.

[9] [9] http://slosh.nws.noaa.gov/sloshPub/SLOSH-Display-Training.pdf

These trends make it necessary and urgent to assess how the city protections need to be upgraded. As a first step, however, an assessment of current risks is required. In the next section, we illustrate how storm surges can be predicted in the current situation by using modelling processes.

2.1 Description of the SLOSH model

SLOSH (Sea, Lake, and Overland Surge from Hurricanes) is a computerized model developed by the American National Weather Service (NWS) with the aim to estimate storm surge heights and winds resulting from historical, hypothetical, or predicted hurricanes[10]. SLOSH is used to define potential flooding from storm surge, for a given location, and from a threatening hurricane.

The SLOSH model contains topographic information for each grid cell. It calculates water surface elevations caused by storm surge in a specific basin and provides output data shown as color-coded storm surge in the SLOSH display (see Figure 1). The water depth indicated in each cell considers the elevation of the grid cell and the amount of water that is able to flow into that cell. For each cell an average water surface elevation is found and assigned to it. Accuracy for SLOSH is usually within +- 20% of the peak storm surge for a known hurricane track, intensity, and size, based on surge measurements from past hurricanes.

A SLOSH Basin is a geographical region where the values of topography, bathymetry, and a hurricane track (considering its pressure, radius of maximum winds, location, direction, and speed) are known. The point of a hurricane's landfall is crucial to determine which areas will be inundated by the storm surge. Data are available for 39 basins in the US.

The model gives as a result different MEOW (Maximum Envelope of Water) which refers to the maximum the water reaches at any point in time at every grid cell in the SLOSH Basin, for a given hypothetical storm. A MEOW is the set of the highest surge values at each grid location for a given storm category, forward speed, and direction of motion and plans for the worst-case scenario. We generated a MEOW for each storm category, storm direction, forward speed, and tide level available for the Bay Biscayne basin[11].

[11] Forward speeds and storm categories were chosen according to shapefiles availability. Not all the categories and forward speeds are provided in a shapefile format.

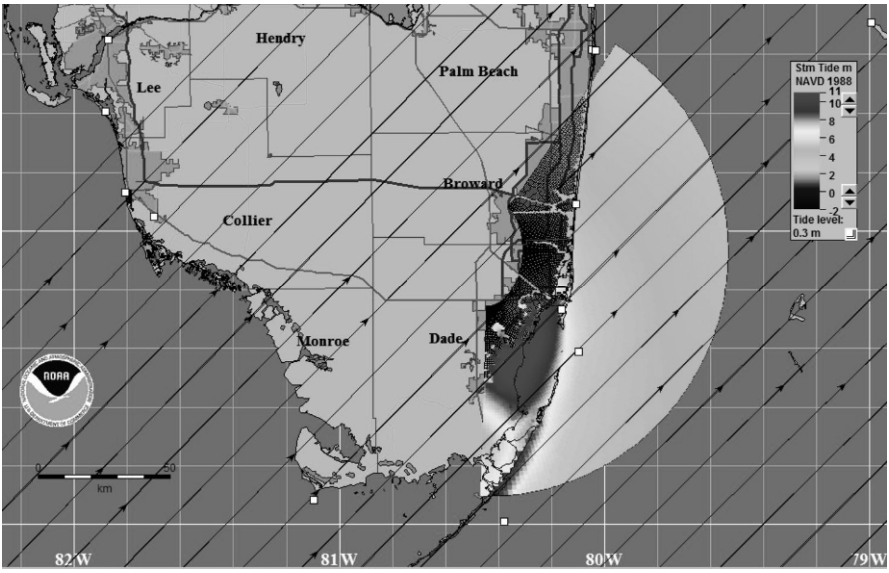


Fig. 1. Category 5 storm heading Northeast at a speed of 25 mph (mean tide) on Biscayne Bay in the SLOSH display.

Local stakeholders and decision-makers in Miami-Dade County are aware of the vulnerability of their territory[12] and they already applied SLOSH in their spatial planning activities. The Miami-Dade County storm surge evacuation zones were redrawn in 2003 following the information acquired through the SLOSH maps: each zone will be evacuated depending on the hurricane's track and projected storm surge.[13]

At present, there is no recognized central authority for climate change risk assessment and adaptation in the Miami metropolitan area. This is due to the USA's decentralization of water management, spatial planning, and related responsibilities.

Therefore, spatial planning and water services are handled by separate agencies. The climate change adaptation effort must engage each municipality and local governmental entity in assessing the impacts of climate on that entity's own responsibility. A multi-stakeholder task force convened by Miami-Dade County has issued preliminary adaptation recommendations and is looking for the collaboration of all local authorities (ICLEI 2009).

The results of our research show that, in the Miami-Dade County area, the Erosion Control and Hurricane Protection Project consists of restora-

[12] http://www.miamidade.gov/derm/climatechange/taskforce.asp
[13] http://www.miamidade.gov/oem/evacuation_zone.asp

tion, ongoing maintenance re-nourishment, and structural improvements of the critically eroded shoreline[14] without taking into account the creation of dams or seawalls.

According to ICLEI (2009), coastal cities and their national governments must not only strengthen their disaster preparedness, such as early warning and evacuation programmes in case of storm events, but also plan ways to handle land development for disaster prevention and to climate proof water. Therefore, both technical innovations and new institutional arrangements are urgently needed.

3 Current flood risks in absence of protection

As a first step, in order to determine flood potential damage in the counties of Miami Dade and Broward, we propose an assessment of the exposure, which is estimated here in absence of flood protection. The exposure is the measure of the values and the assets that would be affected by a flood (Kron 2003). In this analysis, exposure calculation is based on the portion of land that would be inundated in different hypothetical storm surge events.

The available economic data include only insured assets at this stage and were provided to us by RMS. Therefore, infrastructure and government assets are not included at this stage of analysis and will be included in Section 3.2 when analysing flood losses by percentage estimation.

We calculated the exposure considering several possible storm surge simulations (described in Section 2) provided by SLOSH for the Biscayne Bay and integrated in a GIS as shapefile (see Figure 2).

[14] http://www.miamidade.gov/derm/library/erosion_control_budget_plan-10-11.pdf

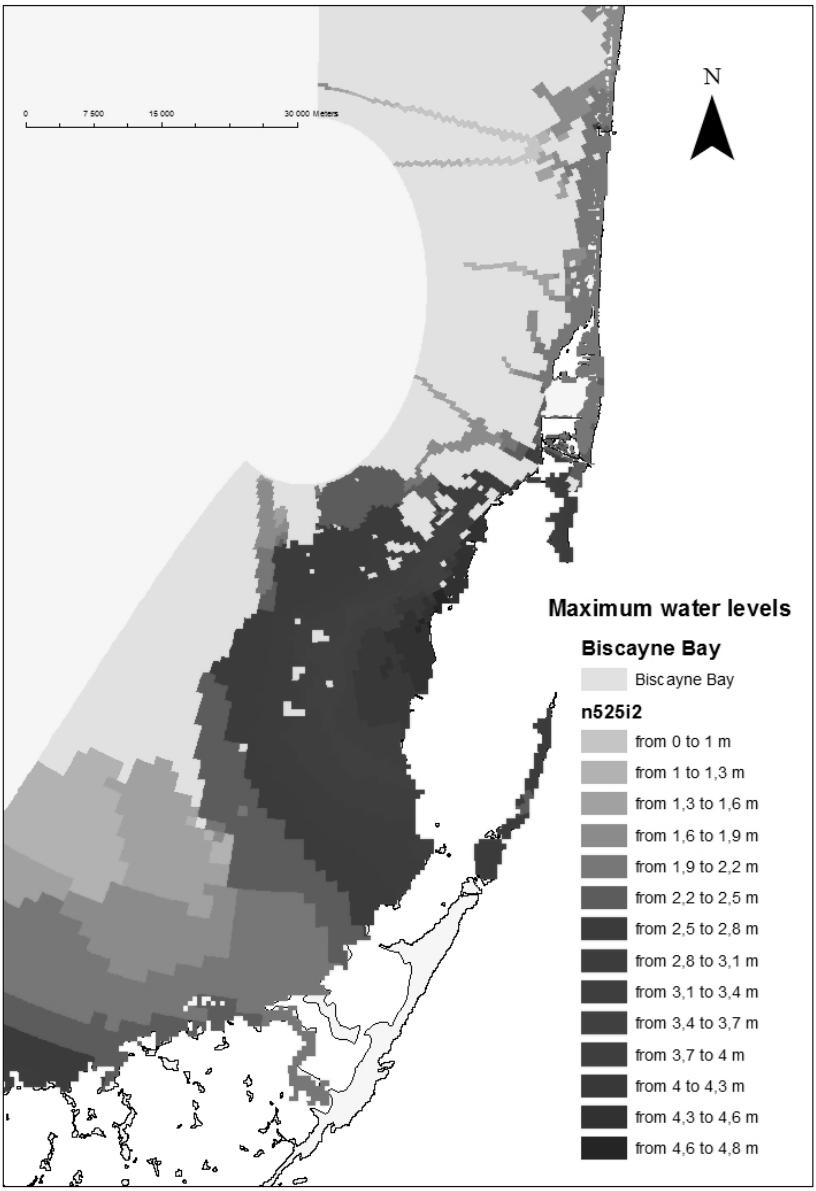


Fig. 2. Maximum water levels in the study area in case of a Category 5 storm heading Northeast at a speed of 25 mph (mean tide).

The SLOSH database exists in this area for different directions of the hurricane tracks: east, north-east, north, north-west, and west. Five hurricane "Categories" (between 1 and 5) and "Speed" (5, 15, and 25 miles per hour) are presented for these directions. We extracted the results in a shapefile format for different storm directions in order to assess the economic losses in case of weak (Category 1), medium (Category 3), or strong (Category 5) storm surge. Most MEOWs have the option of selecting high or mean tide. According to SLOSH training guidelines, that affirm that studies generally use the high tide option, we only considered the high tide results.[15] Therefore we analysed all the high tide MEOWS for the above-mentioned categories and here present the results and analysis for some of the most significant ones[16].

We found the maximum levels that water can reach during the different events in each area. In the Biscayne Bay region, where floods are the largest, maximum water levels from 1 up to 2 meters can be reached in case of a Category 1 storm (depending on the wind direction), 2m up to 3m for a Category 3, and even from 3.5m up to 5m in the extreme event of a Category 5 storm.

By an overlay of these results and the insured value of residential, industrial and commercial areas visualized on a GIS, we determined which current insured built-up areas are at risk of storm surge and estimated the economic insured value of the entire assets that are flooded by each storm surge category. These results are based on a very detailed economic database with a territorial scale of 100 and 500 square meters for the coastal area, and of 1 and 5 km for the hinterland. The economic results we obtained were weighted on a damage function related to water heights, as explained in the next section.

3.1 Damage function

There is a complex link between exposure to high sea level and the destruction and losses caused by such episodes. First, a building that is affected by a flood is not 100-percent destroyed. Thus, direct losses caused by an event have to consider a damage function where losses increase proportionally to water level. Due to the lack of information and the difficulty

[15] http://slosh.nws.noaa.gov/sloshPub/SLOSH-Display-Training.pdf

[16] The first group of letters in the MEOW file name refers to storm motion direction, the next number represents the hurricane category, the next 2 numbers represent the forward speed, I0 refers to mean tide, I2 refers to high tide, and the file extension represents the basin.

in integrating such variables, damage is generally related to only water depth (see for example Green 2003; Van der Sande et al. 2003; Genovese 2006). This basic methodology was outlined already in 1945 by White and is referred to as stage damage curve representing the relation between inundation depth and damage cost for a land use class.

The damage functions are increasing functions, which means that as the inundation depth grows, also damage rises. This value is based on the principle of replacement value: how much money it would cost to obtain the 'identical' object. The damage function has values included between 0 and 1, with the value 0 if there is no damage and the value 1 if there is complete destruction of the asset. Stage-damage curves can be developed from actual flood events and then can be used to simulate damage for potential future events, even though this approach creates problems like extrapolation difficulties from place to place due to differences in warning time and in building type and content (Smith 1994).

Moreover, for storm surge, normally at least two vulnerability curves exist. For properties on the sea front, they will be more quickly destroyed as they will be exposed to wave action as well as flood waters - i.e. the force of the waves will damage the property. This is relevant for Miami since many expensive properties and hotels are located on the sea front. Properties inland will just be exposed to "resting" water damage. This is clearly represented in SLOSH results and fits well with our database of insured properties, where higher values are located in the beach area.

Furthermore, the heights of buildings have to be considered while choosing the damage function. In our study area, both small residential properties and skyscrapers are present; therefore, even if they would require separated vulnerability curves, we chose to use an average curve directly, in order to account for heterogeneity in the results.

We consider here the direct costs, which refer to physical damage to capital assets and inventories, valued at same-standard replacement costs. Indirect losses include those that are not provoked by the disaster itself, but by its consequences (Hallegatte and Przyluski 2010). At this stage, we do not consider indirect losses, such as business interruption, environmental damage, cleaning, and evacuation costs.

Also, only water level effects are considered, even if in case of storm other events can affect the properties, for example strong wind can damage houses' roofs.

During a flood event, some losses can be avoided by appropriate action from the people who live in the floodplain. Examples are the caravans and the cars, because usually there is enough time to remove them from the area that is going to be flooded. Therefore they are not taken into account of the damage assessment. An important question in damage calculation is

which assumption has to be made with respect to the behaviour of the population. This is caused by the fact that damage is a function of many physical and behavioural factors, like the preparedness of a rapid and adequate response to a flood event (Genovese 2006). Hence, all uncertainties in the damage functions are not included in this analysis.

3.2 Damage function application

Among the damage functions available in the literature, we chose the one developed by the OECD for the area of Copenhagen (Hallegatte et al. 2008) because it considers water level until 5 meters (see Table 2) when the others existing in literature for coastal floods consider lower water levels. Of course the Miami area has peculiarities which would require a specific damage function, which we will develop in a following stage of the study. Since in Miami and Miami Beach skyscrapers are numerous and the average building height is probably higher than in Copenhagen, we assume that they will not be completely destroyed during a surge.

Table 2. Damage function for residential, commercial, and industrial structures (Hallegatte et al., 2008). As the inundation depth grows, the damage percentage rises.

Elevation Range (m)	Residential (Structure) %	Commercial (Structure) %	Industrial (Structure) %	Residential (Content) %	Commercial (Content) %	Industrial (Content) %
0	0	0	0	0	0	0
0,5	10	24	20	40	33	38
1	12	40	40	48	55	67
1,5	14	47	47	49	64	75
2	15	54	53	50	73	82
2,5	17	56	55	58	78	85
3	18	58	57	67	82	88
3,5	20	60	59	75	87	91
4	22	61	61	83	91	94
4,5	23	63	63	92	96	97
5	25	65	65	100	100	100

Moreover, we considered the maximum level that water reaches at every grid cell. Therefore, the results we obtained by using this damage function are probably overestimated.

Buildings were distributed in insurance classes, each with their own stage damage curve based on the type of asset (residential, commercial,

and industrial). The contents and building costs have to be calculated separately since their vulnerability to floods are different.

For most residential buildings the most expensive contents are kitchens/heating systems and these assets are most of the time on the ground floor, so are quickly destroyed. For commercial and industrial activities, the difference is even higher since most of these activities are located on ground floors. Therefore, in the damage function, contents are evaluated as completely destroyed at water level that is considerably lower than the buildings themselves.

In order to fit this function to our study, it has been linearly interpolated: values for each 0.10 meters of water level were calculated and extended to 4.8 meters, which is the highest water level that can be potentially reached in case of a Category 5 storm with a north direction.

For each available MEOW, we calculated the total economic damage for different storms of Categories 1, 3, and 5. In Figure 3, we show content and structure damage estimations for storm surges of east and north-east directions, calculated for a low category and forward speed (categories 1 and 0,5 mph speed), for a medium category (3) and 15 mph speed, and for the highest hurricane Category 5 and 15 mph speed.

The estimated direct losses amount to several billions of USD. In the first example, we illustrate that storms having aneasterly direction, in the current economic and land use situation, would cause direct losses to buildings amounting to about 2 to 5 billion USD for residential structures, 5 to 8 billion USD for commercial structures, and 2 to 3 billion USD for industrial structures (depending on storm category). Similarly for the contents, it would cost about 5 to 9 billion USD for residential contents, 4 to 6 billion USD for commercial contents, and 4 to 5 billion USD for industrial contents, for a total of 21 to 35 billion USD.

In the second example, for a storm with a north-east direction, the monetary results are a bit higher, especially for residential structures. The total sum of these results is enormous and shows that, without protection, storm surge increases flooding risks in a significant manner.

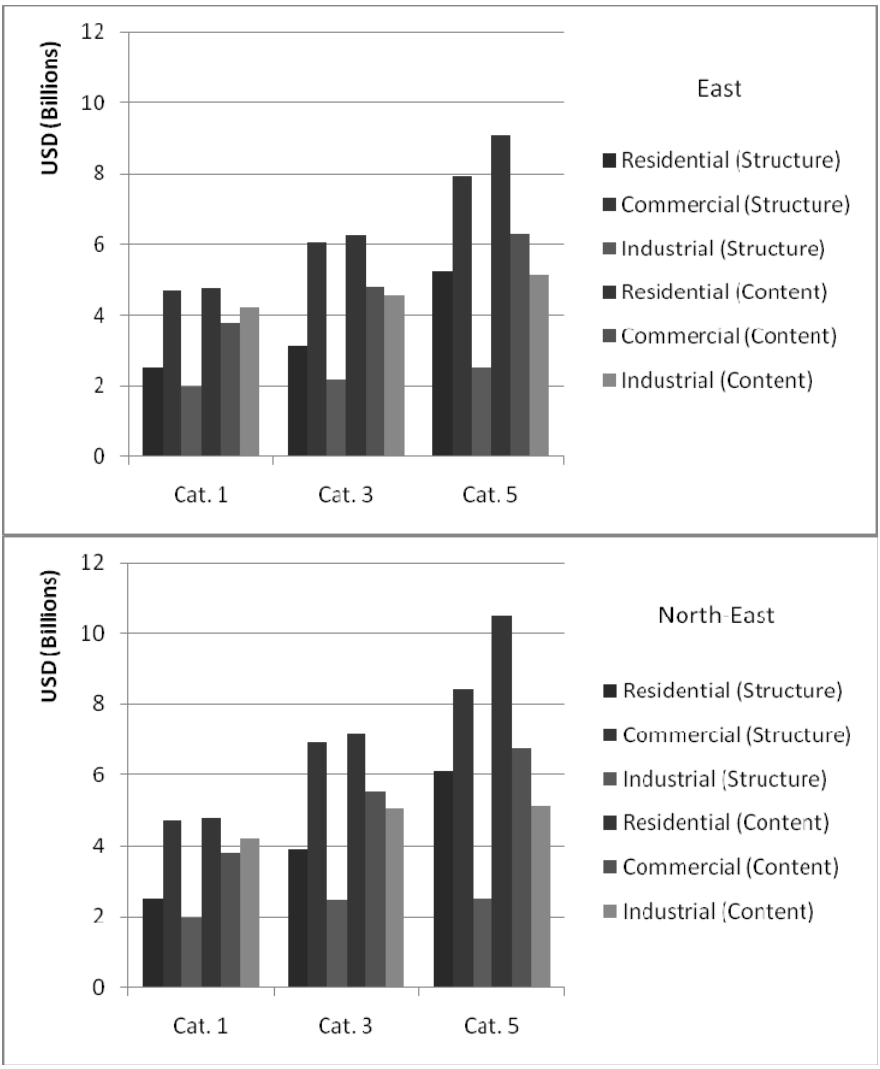


Fig. 3. Direct damage (USD) estimation in Biscayne Bay for storm surge heading east (top panel) and north-east (lower panel), calculated for Categories 1, 3, and 5 for insured contents and structures.

3.3 Non insured assets

In the US and in many other countries, people can insure themselves against flooding and therefore, the potential flood damage is of great interest to insurance companies. These companies have thus created databases

for insurable assets. As mentioned in Section 1, non-insurable assets, such as public infrastructure, are not included in the available data. However, to provide a balanced flood risk estimate, it is essential to include these properties. Since very little information is available on these assets, we refer to past studies in order to find a best guess estimates. Therefore, we used the well documented consequences of the Katrina landfall in New Orleans to help assess infrastructure losses, such as in the OECD report (Hallegatte et al. 2008).

The OEDC report concludes that uninsured losses (infrastructure plus state facilities) represent about 40% of insured losses (residential houses and private properties plus business and commercial activities). Therefore, all the previous outcomes had to be increased by 40%, leading to even larger losses. For a storm with an east direction, losses (building plus contents) are between a minimum loss of 30 billion USD for a Category 1 storm (e105i2) and 50 billion USD for a Category 5 storm (n525i2) (see Figure 4).

The worst-case scenario that we can hypothesize is a Category 5 storm with a north direction and with 25 mph forward speed, which leads to total losses of 118 billion USD. Because of the damage function we chose (see Section 3.1), we assume that our damage evaluation is probably overestimated, especially when considering the areas on the beach front (where most of the buildings are skyscrapers).

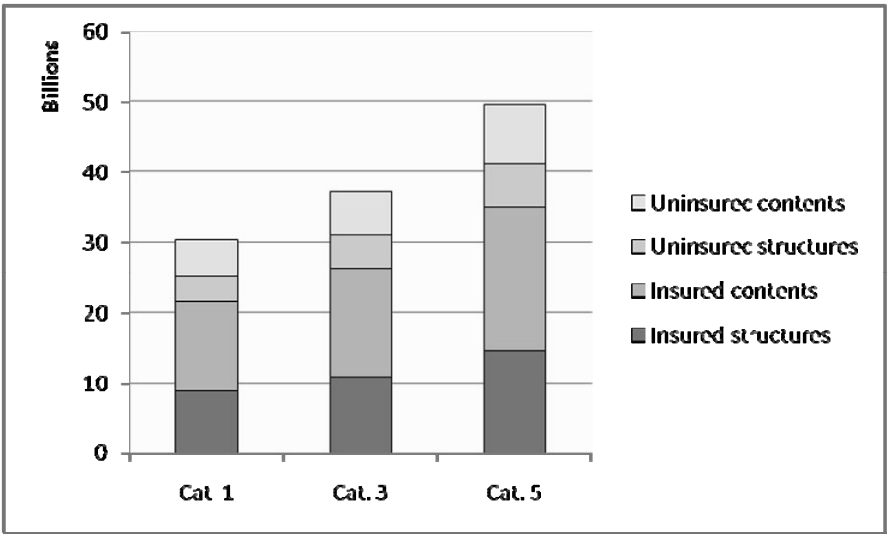


Fig. 4. Insured and uninsured losses for Category 1, 3, and 5 hurricanes, heading east.

4 Assessing risk reduction measures

The previous analysis provided estimates for potential losses and exposure, information which is required to design optimal flood protection through cost-benefit analysis or risk management strategies. The final step of our analysis will be the evaluation of the potential damage when hypothetical protections are built in order to evaluate the benefits from dams and technical defence in the area.

There are three main kinds of vertical shoreline walls used as a protection from storm surges and high tides: seawalls, bulkheads, and revetments. The differences between the three are in their protective function. Seawalls are designed to resist the forces of storm waves; bulkheads are to retain the fill; and revetments are to protect the shoreline against the erosion caused by light waves (U.S. Army Corps of Engineers 1984).

The counties of Miami Dade and Broward have a long coastline that needs to be protected. According to a study of the Pacific Institute on California (Heberger et al. 2000), we can theorize that the cost of building a seawall can be of approximately 1600 USD per meter (in year 2000). A new levee between 3 and 5 meters in height would cost about 460 USD per meter. We can therefore estimate that about 200 km long coast will need to be protected and therefore the cost of constructing a coastal flood protection can be lower than 1 billion USD. In a full cost-benefit analysis of a protection system, monetary costs are not the only costs that need to be taken into account. The visual and physical impacts of protections on the beach also need to be considered because they can make the area less attractive with consequences on economic activities (e.g., tourism) and on quality of life and amenities. In addition, negative consequences on biodiversity and ecosystems are likely.

A full analysis of Miami protection would thus require (i) carrying out a detailed analysis of non-monetary costs of protection infrastructure; and (ii) the consideration of alternative protection measures, in particular, ecosystem-based protection. The current protection policy, based on beach nourishment, goes in this direction, but – as will be shown below – can hardly protect the city against the largest storms.

We made four different basic assumptions hypothesizing different scenarios of intervention: doing nothing, building 2-meter-high dikes, building 3.5-meter-high dikes, and building 5-meter-high dikes to completely protect the area from flood losses.

Unchanged protection: In the current situation, a storm surge, in absence of protection, will lead to losses between 30 billion up to 118 billion

USD for a Category 5 storm, as described in the previous section. It can be assumed that some natural or artificial protections do exist in the area, even if we do not have information about their size and protection capacity. Therefore, this result has to be considered as an overestimation.

2-meter protections: It is difficult to assess the consequences of protection overtopping. Some protection would collapse in case of overtopping, while others are able to support overtopping and keep reducing the water flow within the protected area.

In this analysis, we apply a strong simplification and we assume (i) that all the areas with water levels below 2 meters are not flooded thanks to the protection; and (ii) that in areas with water levels beyond 2 meters, the water level is reduced by 2 meters thanks to the protection. So, where water levels in absence of protection are 5 meters, the protection reduces the flood to 3 meters. This is an optimistic assumption since we suppose that protections remain partly effective in case of overtopping.

A 2meter dam would completely protect from all Category 1 storm surges. A storm surge of Category 3 heading east has a residual damage of 10 and 8% per structure and content of residential building and between 16 and 1% for the commercial and industrial ones (compared with losses in absence of protection). A Category 3 heading north-east will have a residual damage of 25% for residential structure and of 26 and 13% for the commercial and industrial ones.

For a Category 5 storm surge, the 2meter protection is not completely helpful since the percentage of damage being above 2 meters corresponds to 27% for a storm heading east, with a residual loss of about 10 billion USD, and 31% for a storm heading north-east, with a residual loss of 12 billion USD. A storm surge of Category 5 heading north has a residual damage of 66% and the potential loss is 56 billion USD. Moreover, in these circumstances a protection collapse is also possible, since the protections will be overtopped. Therefore, a 2-meter protection could be a suitable protection in case of smaller surges, for example to spare the 22 billion USD of damages caused by a Category 1 storm heading east, but it does not offer an adequate protection for events of major dimension.

3.5-meter protections: A 3.5-meter dam would completely protect from all Category 1 and Category 3 storm surges. Considering Category 5 storms and once again the previous examples, we calculated that, with a protection of 3.5 meters, the flood risk for storm surges heading east and north-east is completely removed. The residual damage is still quite high in case of storm surges of Category 5 heading north, north/north-east, north-west, west, west/north-west and west/south-west. Each of them still

have a residual damage of 9 to 11% and the potential residual loss is between 8 and 11 billion USD.

This partial overflowing could possibly be contained with further flood control structures and defences (e.g., successive dike rings), drainage infrastructure, and beach nourishment interventions.

Figure 5 shows that most of the damage, both in structure and contents, is distributed before the 2- and 3.5- meter level, in the case of a Category 5 heading north hurricane.

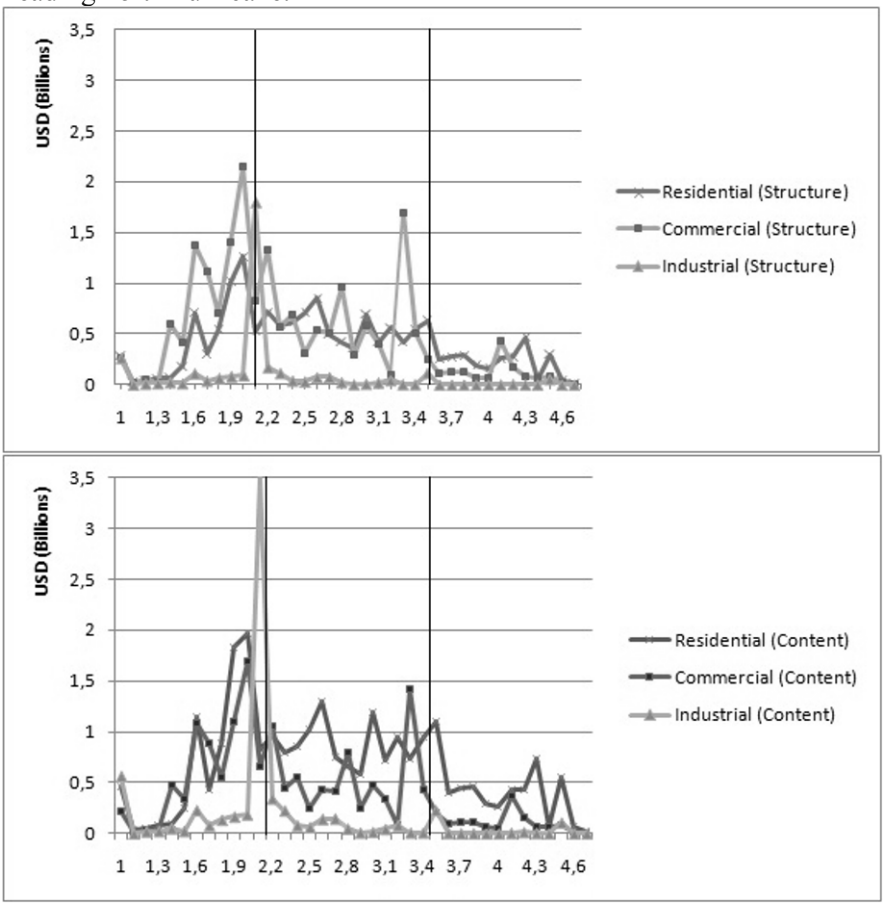


Fig. 5. Economic damage in each flooded area at a given water level caused by a Category 5 heading north hurricane, for structures (top panel) and contents (lower panel).

Completely removing flood risk (5-meter protections): In case of a Category 5 storm, water levels reach levels of about 5 meters and the protection to cope with these events would need to be extremely high and ro-

bust. Even though a full cost-benefit analysis of such a protection is out of the scope of this paper, protecting Miami against all possible storms would probably be extremely expensive, especially because of non-monetary costs, in particular the welfare cost of living behind high walls.

In fact, very high dams would completely eliminate the visual and physical access to beaches. Moreover, in general a hardfill dam requires a basement which is three times the height of the dam itself (ICOLD 1992). This means that a very high dam would require an enormous quantity of space on the beach. On one hand, this solution appears not to be conceivable in an area where tourist attractiveness is the basis of the local economic system. Tourism is the first economic sector in the state and 1.3 million Florida jobs are directly or indirectly related to tourism. The sector – and thus the rest of Florida's economy– is at risk of risk perception shifts due to large disaster. Therefore the impact of protections on the tourist sector can be twofold and has to be carefully investigated.

5 Conclusions

This analysis uses the SLOSH storm surge model and suggests a methodology for assessing direct flood damage potential using a land use database combined with flood extent, flood depth, and economic asset data. We calculated that, in the case of a Category 5 hurricane (as illustrated in Figure 1), water levels can reach about 5 meters in height and potential losses larger than one hundred billion USD and this is without taking into account wind damages. Thus, additional protection seems desirable, even though protecting against all possible events appear simply impossible.

Regardless of their height, it is important to mention that coastal flood defences should not consist only of dams. In Section 1, we showed that beach nourishment interventions are already taking place. There are other options, including: elevating existing areas, building sea walls and flood control structures, and encouraging relocation (Harrington and Walton 2008). Moreover the presence of dams and sea walls requires efficient drainage infrastructure to prevent the city from being flooded by heavy rainfall and surges. In particular, in the presence of high dams, a move from gravity drainage to pumps may be necessary. As a result, protection against storm surge risks must be made in conjunction with improved rainfall flood management.

Furthermore, additional market and non-market impacts of coastal protections should be taken into account while calculating protection costs. Market impacts include the functioning of the harbour, dam maintenance,

drainage, and pumping infrastructures, while non-market impacts include aesthetic considerations and city attractiveness (Hallegatte et al. 2008). In the case of large dikes, these costs may become considerable and will need to be weighed against the benefits of higher protection. Of course, building dams on the beach front may have negative aesthetic effects and may potentially impact city attractiveness and consequently the tourism industry.

Even once appropriate protective measures are built, protections have to be maintained rigorously, since the consequences of a failure or overflowing would be very large. It also highlights the need to adopt emergency plans and warning systems to avoid large human casualties in case of failure. Flood defence upgrades and innovations appear urgently needed in the current context; climate change and sea level rise will make them even more warranted.

Additionally, the design of future protection has to take into account future sea level rise projections due to climate change. Considering the uncertainty of future sea levels and flood risk, adaptation to climate change and to storm surge flood prevention have to be designed together.

It will also be important to build defences in a way that allows for flexibility taking into account the uncertainties in projections and making it possible to upgrade them if sea level rise is larger than expected. In particular, all planning and new infrastructure investments must take account of the risk over the entire lifetime of the investment to reduce unnecessary capital replacement costs.

The present analysis has several caveats which have to be highlighted when considering these results. The assessment of economic impacts associated with coastal flooding has been simplified in several ways. In particular, the damage function has not specifically built for this region. Flood defences have not been explicitly modelled and the consequences of an overflow are not represented in any detail. Flood risks are very different depending on whether an overtopping leads to defence collapse or not. Also, there is large uncertainty concerning damages to infrastructure and other uninsured properties. Most importantly, indirect losses (e.g., business interruption, economic feedbacks) are not included in this analysis, which also disregards important dimensions of social well-being (e.g. casualties, illness, psychological trauma, disruption of social networks, loss of national competitive strength and market positions, loss of cultural heritage, city attractiveness, etc.).

We do not know how population and assets will evolve in Miami over this century. Further studies are necessary to determine how and according to which trends people and buildings will be located in the future. Depending on urbanisation plans and land-use regulations, more buildings can translate or not into a higher exposure. As a consequence, much more

work on the vulnerability of Miami is needed and will be carried out in a follow-up study.

Acknowledgements

We would like to thank Auguste Boissonnade and Robert Muir-Wood from RMS and Edida Rajesh from RMSI for providing us the economic data on asset exposure, and Nicola Ranger from LSE for her advices on vulnerability curves.

References

Emanuel K.A. (2008). "Hurricanes and Global Warming: Results from Downscaling IPCC AR4 Simulations", Bulletin of the American Meteorological Society.

Genovese E. (2006). A methodological approach to land use-based flood damage assessment in urban areas: Prague case study, Technical EUR Reports, EUR 22497 EN.

Green C. (2003). Handbook of Water Economics: Principles and Practice, John Wiley and sons, Chicester, 443 pp.

Hallegatte, S., J.-C. Hourcade, and P. Dumas (2007). Why economic dynamics matter in assessing climate change damages: illustration on extreme events, Ecological Economics, 62, 330–340.

Hallegatte S., N. Patmore, O. Mestre, P. Dumas, J. Corfee Morlot, C. Herweijer, R. Muir Wood (2008). Assessing Climate Change Impacts, Sea Level Rise and Storm Surge Risk in Port Cities: A Case Study on Copenhagen, OECD Environment Working Paper No. 3, 2008 (2).

Hallegatte S. and V. Przyluski (2010). The economics of natural disaster, CESifo Forum 2/2010, pp. 14—24.

Harrington J., and Walton T. L. (2008). Climate change in coastal area in Florida: sea level rise estimation and economic analysis to year 2080, Florida State University report.

Heberger M., H. Cooley, P. Herrera, P. H. Gleick, E. Moore (2009). The impacts of sea-level rise on the California coast, California Climate Change Center, Pacific Institute, August 2009.

Herweijer C, Nicholls R. J., Hanson S, Patmore N, Hallegatte S, Corfee-Morlot J, Chateau J, Muir-Wood R (2008). "How do our coastal cities fare under rising flood risk?", Catastrophe Risk Management, April, 12-13.

ICOLD (1992). Cost Impact on Future Dam Design - Analysis and Proposals, ICOLD publications, No. 83.

ICLEI (2009). Perspectives on water and climate change adaptation. Local government perspective on adapting water management to climate change. http://worldwatercouncil.org/fileadmin/wwc/Library/Publications_and_reports/Climate_Change/PersPap_07._Local_Government.pdf

IPCC (2007). Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change [Solomon S, Qin D., Manning M, Chen Z, Marquis M, Averyt K B, Tignor M, Miller H L (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 996 pp.

Kron W. (2003). High water and floods: resist them or accept them? In: Schadenspiegel (Losses and loss prevention), 46th year. No.3. 26-34. Munich Re Group, Munich.

Landsea C. W. (2005). "Hurricanes and global warming". Nature, 436, 686–688 (2005).

Lugeri N., E. Genovese, C. Lavalle, A. De Roo (2006). Flood risk in Europe: analysis of exposure in 13 Countries, Technical EUR Reports, EUR 22525 EN.

Lugeri N., Kundzewicz Z.W., Genovese E., Hochrainer S., Radziejewski M. (2010). River flood risk and adaptation in Europe: assessment of the present status. Mitigation and Adaptation Strategies for Global Change International Journal, Volume 15, Number 7, 621-639.

Nicholls, R.J., S. Hanson, C. Herweijer, N. Patmore, S. Hallegatte, J. Corfee-Morlot, J. Chateau, R. Muir-Wood (2007). Screening Study: Ranking Port Cities with High Exposure and Vulnerability to Climate Extremes, OECD Working Paper, at http://www.oecd.org/document/56/0,3343,en_2649_201185_39718712_1_1_1_1,00.html

Pfeffer, W. T., Harper, J. T., O'Neel, S., (2008). Kinematic constraints on glacier contributions to 21st-century sea-level rise, Science, 321(5894): 1340-1343

Rahmstorf (2007). Sea-Level Rise A Semi-Empirical Approach to Projecting Future, Science 315, 368–370, DOI: 10.1126/ science.1135456.

Smith, D. I. (1994). Flood damage estimation – A review of urban stage-damage curves and loss functions, Water SA, 20(3), 231–238, 1994.

United States Army Corps of Engineers (1984). Shore Protection Manual, Volumes 1&2. Waterways Experiment Station, Coastal Engineering Research Center, Vicksburg, Mississippi.

Van der Sande C.J., de Jong S.M., de Roo A.P.J. (2003). A segmentation and classification approach of IKONOS-2 imagery for land cover mapping to assist flood risk and flood damage assessment, International Journal of Applied Earth Observation and Geoinformation, pp. 217–229.

Webster, P.J., G.J. Holland, J.A. Curry, H.-R. Chang (2005). "Changes in Tropical Cyclone Number, Duration, and Intensity in a Warming Environment", Science, 309, 1844,1846, 16 September 2005.

White G.F. (1945). "Human Adjustments to Floods: A Geographical approach to the Flood Problem in the United States", Doctoral Dissertation and Research paper no. 29, Department of Geography, University of Chicago.

Mining Sequential Patterns from MODIS Time Series for Cultivated Area Mapping

Yoann Pitarch[1], Elodie Vintrou[2,4], Fadi Badra[3,4], Agnès Bégué[2,4] , Maguelonne Teisseire[3,4]

[1]LIRMM - CNRS - UM2, [2]CIRAD, [3]Cemagref, [4]TETIS, Montpellier, France
pitarch@lirmm.fr, {elodie.vintrou, fadi.badra, agnes.begue, maguelonne.teisseire}@teledetection.fr

Abstract. To predict and respond to famine and other forms of food insecurity, different early warning systems are using remote analyses of crop condition and agricultural production by using satellite-based information. To improve these predictions, a reliable estimation of the cultivated area at a national scale must be carried out. In this study, we developed a data mining methodology for extracting cultivated domain patterns based on their temporal behavior as captured in time-series of moderate resolution remote sensing MODIS images.

1 Introduction

The northern fringe of sub-Saharan Africa is a region considered particularly vulnerable to climate variability and change and food security remains there a major challenge.

One of the preliminary stages necessary for analyzing such impacts on agriculture and food security is a reliable estimation of the cultivated domain at a national level, a scale compatible with climate change studies. For that purpose, different early warning systems such as FEWS and JRC-MARS use global land cover maps but they are generally focused on large ecosystems and are not suitable for fragmented and heterogeneous African landscapes. Recent moderate-resolution sensors, such as MODIS/TERRA,

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_3,

with spatial resolutions as low as 250 m, offer new possibilities in the study of agricultural lands. With this increase in spatial resolution, the detection of groups of fields can now be considered. The low and medium spatial resolutions do not, by themselves, provide a completely satisfactory representation of the landscape but are compensated by a large coverage area and by an excellent temporal resolution.

This brings us to the question whether moderate-resolution satellite data in combination with external data (fields surveys, climate etc.) can provide a correct assessment of the distribution of the cultivated domain at the country level. It is expected that more consistent information on vegetation would allow monitoring Sahelian rural landscapes with better continuity, thereby providing relevant information for early warning systems.

In this study, we develop a data mining methodology to extract relevant sequential patterns to describe cultivated areas. These patterns are obtained from the static description and the temporal behavior as captured in time-series of moderate resolution remote sensing images. We applied this methodology in Mali, a representative country of the Sahel Belt of Africa. Both the temporal and spatial dimensions add substantial complexity to data mining tasks. A prioritization is needed to reduce the search space and to allow the relevant pattern extraction. We thus adopt a two-step approach: (1) identification of relevant descriptors per class; and (2) associated pattern mining from MODIS time series.

2 The data description

2.1 Study area

Mali is, after Senegal, the second westernmost country of West Africa around Latitude 14°N. It displays a South-North climatic gradient that ranges from subtropical to semi-arid and extends further north to desertic. As for other West African countries along the same latitudinal belt, food security relies on an adequate supply of rainfall during monsoon seasons. This country can therefore be considered representative of the Soudano-Sahelian zone, where a strong dependence on rain-fed agriculture implies vulnerability to major changes due to climate and human activities, and hence requires specific attention. Particular attention was paid to three zones in the Bani catchment, mainly located in Southern Mali (Table 1).

Table 1. Main characteristics of the three studied sites

Site name (eco-climatic zone)	Mean annual rainfall	Main crops	Natural vegetation type
Cinzana (Soudano-Sahelian)	600 mm	Millet, sorghum	High proportion of bare soils and sparse vegetation
Koutiala (Soudano-Sahelian)	750 mm	Cotton, millet, sorghum	Large areas of semi-open and closed natural vegetation
Sikasso (Soudanian)	1000 mm	Maize, cotton, fruit crops	Dense natural vegetation

2.2 Data

Field data Field surveys were conducted in Mali during the 2009 and 2010 cropping seasons (from May to November) in order to characterize Soudano-Sahelian rural landscapes. Three sites (Cinzana, Koutiala, and Sikasso) were selected to sample the main agro-climatic regions of Central and Southern Mali (Table 1). 980 GPS waypoints were registered, and farmers were interviewed. Each waypoint was transformed into a polygon whose center has been affected by land use.

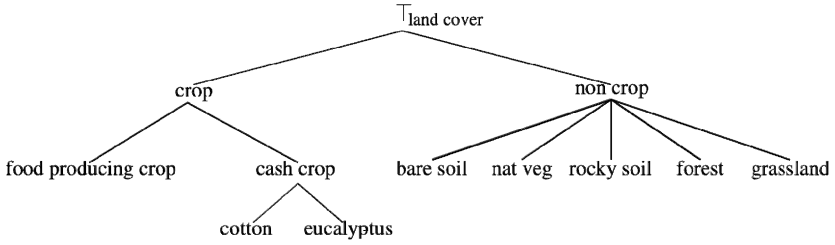


Fig. 1. Crop hierarchy

External data Six static descriptors were also used to characterize the site surveys: soil type, distance to the village, distance to the river, rainfall, ethnic group, and village name. The domains of associated data values are detailed in Table 2.

Images data MODIS time series: The NASA Land Process Distributed Active Archive Center (LP DAAC) is the repository for all MODIS data. Amongst MODIS products, we selected the 'Vegetation Indices 16-Day L3 Global 250 m SIN Grid' temporal syntheses for our study. For Mali, a set of 12 MODIS 16-days composite normalized difference vegetation

index (NDVI) images (MOD13Q1/V05 product) at a resolution of 231.6 m were acquired for 2007 (we keep the best quality composite image out of two for each month).

The year 2007 was chosen to overlap with the more recent high-resolution data available. We assume that the observed classes of land use remained globally unchanged from 2007 to 2009 (field surveys in 2009). However, Malian farmers practice "crop rotation". It is the practice of growing a series of dissimilar types of crops in the same area in sequential seasons for various benefits such as to avoid the buildup of pathogens and pests that often occurs when one species is continuously cropped, improving soil structure and fertility. Thus, we decided to only consider the two higher levels of the crop hierarchy (Figure 1).

Remotely sensed indices used

- *Normalized Difference Vegetation Index:* NDVI is one of the most successful index to simply and quickly identify vegetated areas and their "condition", providing a crude estimate of vegetation health. It displays the relationship between the quantity of chlorophyll in leaves with red and near infrared wavelength, so that the NDVI image is used to search vegetation as estimating biomass, plant productivity, and fractional vegetation cover (Rouse 1974).

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

where RED and NIR stand for the spectral reflectance measurements acquired in the red and near-infrared regions, respectively. In general, NDVI values range from -1.0 to 1.0, with negative values indicating clouds and water, positive values near zero indicating bare soil, and higher positive values of NDVI ranging from sparse vegetation (0.1 - 0.5) to dense green vegetation (0.6 and above). Furthermore, different land covers exhibit distinctive seasonal patterns of NDVI variation. Crops have generally a distinct growing season and period of peak greenness, which allows the discrimination with other types of land cover.

- *Texture:* Information content in a digital image is expressed by the 'intensity' of each pixel (*i.e.* tone or color) and by the spatial arrangement of pixels (*i.e.* texture, shape, and context) in the image. Traditionally, tone (*i.e.* spectral intensity) has been the primary focus for most image analysis and hence information extraction in remote-sensing studies. However, texture analysis is examined as an important contributor to scene information extraction. The majority of image classification procedures, particularly in operational use, relies on spectral 'intensity' characteristics alone and thus is oblivious to the spatial information content of the image.

Textural algorithms, however, attempt to measure image texture by quantifying the distinctive spatial and spectral relationships between neighboring pixels. In response to the need for extracting information based on the spatial arrangement of digital image data, numerous texture algorithms have been developed. Statistical approaches, such as those developed by Haralick et al. (1973) make use of gray-level probability density functions, which generally are computed as the conditional joint probability of pairs of pixel gray levels in a local area of the image. In this study, we used four Haralick textural indices (Haralick, 1979) calculated on the MODIS time series: "variance", "homogeneity", "contrast", and "dissimilarity" on ENVI®. The Haralick textural features describe the spatial distribution of gray values and the frequency of one gray tone appearing with another gray tone in a specified distance and at a specified angle. The generation of these indices is based on different orientations of pixels pairs with specific angle (horizontal, diagonal, vertical, co-diagonal) and distance, called "patterns". We determined empirically a size of pattern of 15 pixels for MODIS, which is the smaller patch repeated in different direction and distance.

Table 2. The domains of external data values

Dimension D_i	Values
Id-pt	{1,2, … ,980}
Date	{1,2, … ,12}
Site name	{cinzana, koutiala, sikasso}
Crop	{millet, sorghum, rice…}
Soil type	{clay loam, sandy loam, gravelly soils…}
Distance to the village	[200 ; 30000]
Distance to the river	[5000 ; 68000]
Rainfall (mm)	{600, 750, 1000}
Ethnic group	{bambara, senoufo, bozo…}
Name of the village	{dioforongo, tigui, sanando…}

3 Motivating Example

In order to illustrate our approach, we consider the following example that will be used throughout the paper. Let us consider a relational table *T* in which *NDVI* values by field are stored. More precisely, we assume that *T* is defined over six dimensions (or attributes) as shown in Table 3 and where: *D* is the date of statements (considering two dates, denoted by 1 and 2); *I* is the field identifier (considering four different fields, denoted by

F1, F2, F3, and F4); *C* is the crop type (considering two discretized values, denoted by FP (food-producing) and NFP (non food-producing)); *S* is the soil type (considering three different soil types, denoted by GS (gravelly soils), SL (sandy loam) and CL (clay loam)); *DV* is the distance between the associated field and the nearest village (considering two discretized values, denoted by near and far); and *NDVI* stands for the NDVI value associated to each field at each timestamp (considering 4 abstract values n_1, n_2, n_3 and n_4).

We consider five sets of dimensions as follows:

(i) the dimension *D* representing the date;
(ii) the dimension *I* representing the identifier;
(iii) the dimensions *S* and *DV*, that we call *static dimensions* or *descriptors* (values of these dimensions associated to a given field do not evolve over time);
(iv) the dimension *NDVI*, that we call *dynamic dimension* or *indicators* (values of these dimensions associated to a given field evolve over time); and
(v) the dimension *C* that we call the class.

For instance, the first element of *T* (Table 3) means that the field 1 is a food-producer crop composed by CL, near to a village, and that at date 1 the NDVI value was n_1. Observing in great details the static attribute values per class, some comments should be made. First, food-producing crops are always located near to the village whereas the soil composition is changing. Similarly, non food-producing crops are always cultivated on GS whereas the distance to the nearest village is changing.

A first interpretation to these comments is that the dimension *DV* appears to be decisive to identify food-producing crops whereas the dimension *S* appears to be decisive to identify non food-producing crops. Consequently, it is pertinent to only consider decisive dimensions per crop to mine representative rules. Once static dimensions have been filtered, the dynamic dimension (*NDVI*) is considered in order to mine sequential patterns characterizing crops.

Let us suppose that we look for sequences that are verified by all the crops in a given class. Under this condition, the pattern $<(near, n_1)(near, n_2)>$ (meaning that fields located near to a village and where the NDVI statement are n_1 at a certain date and n_2 after) characterizes the food-producing crops and the pattern $<(GS, n_3)>$ characterizes the non food-producing crops. It should be noted that representative rules per class could be composed by values of different dimensions. In the rest of this paper, we describe the adopted methodology to determine the decisive attributes per class and how the table *T* is subdivided and mined to obtain representative rules per class.

Table 3. Table *T*

D (Date)	I (Id)	C (Crop)	S (Soil)	D (Distance to village)	NDVI (NDVI value)
1	F1	FP	CL	near	n_1
1	F2	FP	SL	near	n_2
1	F3	NFP	GS	far	n_3
1	F4	NFP	GS	near	n_4
2	F1	FP	CL	near	n_1
2	F2	FP	SL	near	n_2
2	F3	NFP	GS	far	n_3
2	F4	NFP	GS	near	n_4

4 Preliminary Definitions

In this section, concepts and definitions concerning multidimensional sequential patterns are presented and are inspired by the notations introduced in Plantevit et al. (2010). For each table defined on the set of dimensions D, we consider a partition of D into three sets: D_t for the temporal dimension, D_A for the analysis dimensions, and D_R for the reference dimension. Each tuple $c = (d_1, \ldots, d_n)$ can thus be denoted $c = (r, a, t)$ with r the restriction on D_R, a the restriction on D_A and t the restriction on D_t .

Definition 1. *(Multidimensional Item) A multidimensional item e defined on* $D_A = \{D_{i1}, \ldots, D_{im}\}$ *is a tuple* $e = (d_{i1}, \ldots, d_{im})$ *such that* $\forall k \in [1, m], d_{ik} \in Dom(D_{ik})$.

Definition 2. *(Multidimensional Sequence) A multidimensional sequence S defined on* $D_A = \{D_{i1}, \ldots, D_{im}\}$ *is an ordered non-empty list of multidimensional items* $S = < e_1, \ldots, e_l >$ *where* $\forall j \in [1, l]$, e_j *is a multidimensional item defined on* D_A.

Considering our running example and that $D_A = \{DV, NDVI\}$, (near, n_1) is a multidimensional item,
<(near, n_1)(near, n_2)> is a multidimensional sequence on D_A.

Remark In the original framework of sequential patterns (Agrawal and Srikant, 1995), a sequence is defined as an ordered non-empty list of item sets where an item set is a non-empty set of items. Nevertheless, in the scope of this paper, we only consider item sequences since at each date, one and only one item can occur for each field. For instance, only one NDVI statement is available per date and field.

An identifier is said to support a sequence if a set of tuples containing the items satisfying the temporal constraints can be found.

Definition 3. (Inclusion) *An identifier $r \in Dom(D_R)$ supports a sequence $S = < e_1, \dots, e_l >$ if $\forall j \in 1 \dots l, \exists d_j \in Dom(D_t), \in t=(r,e_j,d_j) \in T$ where $d_1 < d_2 < \dots < d_l$.*

Definition 4. *(Sequence Support) Let D_R be the reference dimension and T the table. The support of a sequence S is:*

$$support(S) = \frac{\left| \left\{ r \in Dom(D_R) \, s.t. \, r \text{ supports } S \right\} \right|}{\left| Dom(D_R) \right|}$$

Definition 5. *(Frequent Sequence) Let $minSupp \in [0, 1]$ be the minimum user-defined support value. A sequence S is said to be frequent if $support(S) \leq minSupp$.*

Considering the definitions above, an item can only be retrieved if there exists a frequent tuple of values from domains of D_A containing it. For instance, it can occur that neither *(CL, near)*, *(SL, near)* nor *(GS, near)* is frequent whereas the value *near* is frequent. Thus, Plantevit et al. (2010) introduces the *joker* value *. In this case, we consider *(*, near)* which is said to be *jokerized*.

Definition 6. *(Jokerized Item)*
Let $e = (d_1, \dots, d_m)$ a multidimensional item. We denote by $e[d_i/\delta]$ the replacement in e of d_i by δ. e is said to be a jokerized multidimensional item if: (i) $\forall i \in [1, m], d_i \in Dom(D_i) \cup *$, (ii) $\in i \in [1, m]$ such that $d_i \neq *$ and (iii) $\forall d_i = *, \in \delta \in Dom(D_i)$ such that $e[d_i/\delta]$ is frequent.

A *jokerized* item contains at least one specified analysis dimension. It contains a * only if no specific value from the domain can be set. A *jokerized* sequence is a sequence containing at least one *jokerized* item.

5 Method

5.1 Overview

In this paper, we aim at discovering representative rules in order to characterize crop classes and propose a four-step method to achieve this issue. It should be noticed that the crop classes depends on the user-defined interest level of the crop hierarchy displayed in Figure 1. For instance, assuming that the user would like to discover representative rules for classes in the second level of the hierarchy, the set of classes will be {*food-producing, non food-producing, other*}. Such rules could be specific to one site or corresponding to all sites.

These four steps are illustrated in Figure 2 and are briefly presented here:

- *The raw database pretreatment.* During this phase, two actions are performed. First, since the raw database stores crops at the lowest level of the hierarchy, these attribute values must be rewritten to match with the user-defined interest level. Second, sequential pattern mining aims at discovering frequent relations in a database but is not well adapted to mine numerical attributes (*e.g.*, distance to the village, NDVI value) due to the huge domain of definition for such attributes. Consequently, numerical attributes are discretized to improve the sequential pattern mining phase.
- *The build of projected databases.* Since we would like to obtain representative rules per class, the pretreated database is projected on the different class values.
- *The decisive attribute computation.* During this step, a search is performed on each projected databases in order to find and delete non-decisive static attribute dimensions. Intuitively, a static attribute is said to be non-decisive if none of its values allows characterizing the class. More precisely, we guarantee that if any value of a static attribute appearing in at least *minSupp%* does not exist in the projected database, the representative rules associated to this class will never contain specific values of this static attribute. Consequently, it is useless to consider it in the rest of the process and this attribute will be removed from the projected database
- *The sequential pattern mining.* Once the projected databases were cleaned up, the algorithm M^2SP is applied to each database. We obtain a set of frequent patterns for each class.

These steps are now detailed in the following subsections.

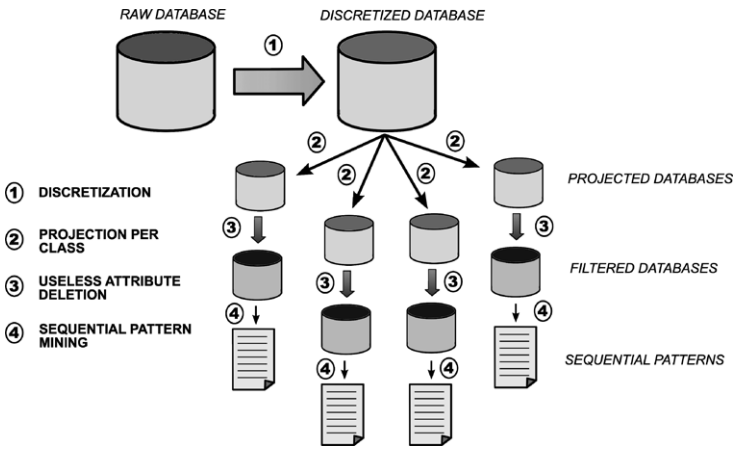


Fig. 2. Overall schema of the proposed methodology

5.2 The Database Pretreatment and Projections

The first performed treatment is the rewriting of the database in order to make the crop attribute values and the user-defined interest level match. This is motivated by two reasons. First, as mentioned in Section 2, mining representative rules for precise crop values is not consistent. As a consequence, crop attribute values must be rewritten to, at least, the above level of granularity. Second, since the hierarchy is composed of two workable levels of granularity, it is interesting to allow the user to choose which level must be explored. Consequently, a user-defined parameter, *Level*, is introduced to specify which level of granularity to mine. Thus, rules representing different generalized classes can be compared. An illustration of this database rewriting is displayed in Table 3, where crop attribute values have been already generalized to the second level of granularity (*i.e.*, *Dom(Crop)= {FP,NFP}*).

A second pretreatment is the discretization of numerical attributes. This discretization is motivated by the use of the sequential pattern technique to mine representative rules. Indeed, sequential pattern algorithms aim at discovering frequent relations among the fields belonging to the same class. When dealing with numerical attributes, two values can be considered as different items even if they are very close. For instance, let us consider that the distance to the nearest village is 200 m for field 1 and 205 m for the field 2. These two distances would have been considered different items by the M^2SP algorithm without discretization even if they are semantically

closed. In our application case, numerous attributes are numerical. Thus, this discretization is necessary. Numerous discretization techniques can be found in the literature (Catlett 1991). Section 6 details the adopted technique per numerical attribute.

Once the database was pretreated, projections per crop attribute values are performed. Indeed, this is motivated by the fact that we would like to discover representative rules per class. Thus, an intuitive way to achieve this goal is to subdivide the pretreated table into smaller ones associated to each class. Regarding our running example, Tables 4 and 5 display the result of this projection.

Table 4. T_{FP}, the *FP* projected table

D (Date)	I (Id)	S (Soil)	D (Distance to village)	NDVI (NDVI value)
1	F1	CL	near	n_1
1	F2	SL	near	n_1
2	F1	CL	near	n_2
2	F2	SL	near	n_2

5.3 Dimensionality Reduction

Once the projected databases were built, a search is performed on the static attributes of each database in order to identify useless static attributes. Intuitively, if values of a static attribute are very changing, this attribute is not really characteristic to this class. So, it can be deleted from the projected class. The main advantage of such a strategy is to reduce the search space during the sequential pattern mining phase.

Indeed, it is empirically shown in Plantevit et al. (2010) that the number of dimensions exponentially impacts on both the memory consumption and the extraction time. Whereas traditional applications domains often deal with few analysis dimensions, this point can be very problematic in our context since the number of both static and dynamic dimensions can be high. For instance, experiment results presented in Section 6 concern at most 12 dimensions. Traditional multidimensional sequential pattern approaches cannot efficiently deal with such a number of analysis dimensions.

Moreover, independent of performance considerations it is important to notice that the higher the number of dimensions, the higher the number of

extracted patterns. Since experts will exploit these extracted patterns, reducing the dimensionality without loss of expressivity is very relevant to improve the result analysis phase.

Table 5. T_{NFP}, the *NFP* projected table

D (Date)	I (Id)	S (Soil)	D (Distance to village)	NDVI (NDVI value)
1	F3	GS	far	n_2
1	F4	GS	near	n_3
2	F3	GS	far	n_4
2	F4	GS	near	n_3

To perform such a dimensionality reduction, we proceed as follows: Let *minSupp* be the user-defined parameter used during the sequential pattern-mining phase, T_i be a projected database and $D_j \in T_i$ be the static dimension in T_i. It can be easily proved that if any value of D_j appearing in at least *minSupp* * | T_i | tuples in T_i (where | T_i | is the size of T_i) does not exist, any sequential pattern extracted from T_i where a value of D_j appears cannot exist. If so, the dimension D_j is considered as useless and is thus deleted from T_i.

A direct corollary of this property is that if an attribute is retained, at least one sequential pattern containing a value of D_j. will exist To illustrate this affirmation, let us consider, T_{FP}, the projected database presented Table 4 and *minSupp* = *1*. The two static attributes are *DV* and *S*. Regarding the *DV* attribute, all the tuples share the same value (*near*). This attribute is considered as useful for the next step and is thus retained.

Let us now consider the *S* attribute. Here, no value satisfies the *minSupp* condition. As a consequence, *S* is deleted from this table. To attest the consequence of such a strategy, let us consider SP_{FP}, the set of the multidimensional sequential patterns extracted from T_{FP} where *minSupp* =*1*, D_t=*D*, D_R= *I* and D_A={*C, S, NDVI*} (*i.e.*, all the static and dynamic attributes are considered). Under these conditions, SP_{FP}= {<(*,near, n_1)>, <(*,near,n_1)(*,near,n_2)>}. It is readily noticeable that *DV* occurs in SP_{FP} but not *S*.

It is interesting to observe that the set of useful attributes per class can be different. As a consequence, independent of the values of these attributes, attributes themselves can be representative of one class. For instance, performing the above described dimensionality reduction technique on T_{NFP} (see Table 5), *S* but not *DV* will be retained this time.

5.4 Mining Representative Rules

Once useless attributes have been deleted, the M^2SP algorithm is applied on each projected and cleaned database T_i such that *minSupp* is defined the same as during the previous step, D_t=D, D_R= I and D_A is composed of the retained static attributes and the dynamic attributes. We note SP_{Ti} is the set of sequential patterns extracted from T_i. For instance considering T_{FP} and *minSupp* = 1 <(near,n_1)(near,n_2)> is a frequent sequence meaning that NDVI values equal n_1 and then n_2 is a frequent behavior for fields cultivating food-producing crops located near a village.

6 Experiment Study

In this section, we present experiments to evaluate the feasibility and efficiency of our approach. Throughout the experiments, we answer the following questions inherent to efficiency issues: *Does the dimensionality reduction technique allow deleting useless static attributes without loss of information? Does the mining process allow discovering discriminating patterns per class? Does the texture data allow a better discriminating pattern extraction than only considering NDVI values?* The experiments were performed on Intel(R) Xeon(R) CPU E5450 @ 3.00GHz with 2GB of main memory, running Ubuntu 9.04. The methods were written in Java 1.6. We first describe the adopted protocol and then present and discuss our results.

6.1 Protocol

The method was evaluated on the dataset described in Section 2. This dataset contains 980 distinct fields and a MODIS time series of length 12 is associated to each field. The 7 static dimensions and the 5 dynamic dimensions were the same as described in Section 2.

As mentioned in Section 5, a discretization step is necessary to efficiently mine frequent patterns. The adopted discretization methods are as follows:

- EQUI-WIDTH technique (the generated intervals have the same width) was used for distance village and distance river attributes
- EQUI-DEPTH technique (the generated intervals have the same size) was used for the other numerical attributes.

In this experiment study, two sets of classes were considered. The first set of classes, denoted by B aims at discovering patterns allowing the dis-

tinction between food-producing crops (FP), non food-producing crops (NFP), and non crops (OTHER). The second set of classes, denoted by *C*, aims at discovering patterns allowing the distinction of more general classes: crops (*Cr*) and non crops (*NCr*).

In order to evaluate the impact of texture data in discriminating pattern extraction, we consider a first configuration, denoted by *Default*, where all the dynamic attributes were used. On the contrary, the configuration denoted by *NDVI* is only composed of NDVI values as a dynamic attribute.
Three experimental results are presented and discussed in this section:

1. A first experiment was performed to evaluate the number of retained static attributes according to two *minSupp* values.
2. A second experiment was performed to evaluate the number of discriminating patterns. Here, *discriminating* means that a pattern appears in one class but not in the others.
3. Finally, the last experiment was performed to observe the discriminating dimension values according the two above described configurations.

6.2 Results and Discussion

Table 6 displays the retained attributes according to the two sets of classes and two *minSupp* values. First of all, it can be noticed that the *minSupp* threshold value has an obvious impact on this attribute selection.

Indeed, considering *minSupp=0.5*, more than half of the attributes were deleted. Moreover, it is interesting to observe that the retained attributes per class and set of classes are roughly identical.

Table 7 displays the proportion of discriminating patterns per class with *minSupp=0.5* and the NDVI configuration. Indeed, even if a pattern was extracted from one class, it is not enough to consider it as discriminating (*i.e.*, the same pattern can appear in different classes). Thus, queries was formulated to search which patterns appear in one class and not in the others. Two conclusions can be drawn from this figure. First, considering the set of classes *B*, most of the extracted patterns are discriminating (even if the *FP* class obtains a worse score). Second, finding discriminating patterns on the set of classes *is* more difficult.

Table 6. Retained static attributes under default configuration

Level	Class	Static attributes (*minSupp=0.5)*						Static attributes (*minSupp=0.3)*						
		Distance village	Site name	Ethnic group	Rainfall	Soil type	Distance river	Village Distance	village	Site name	Ethnic group	Rainfall	Soil type	Distance river	Village
B	FP	x	x	x	x	x			x	x	x				
	NFP	x	x	x	x	x			x	x					
	OTHER	x	x	x	x	x			x	x					
C	Cr	x	x	x	x	x			x	x					
	NCr	x	x	x	x	x			x	x					

Table 7. Proportion of discriminating patterns per class with *minSupp=0.5* and the NDVI configuration

Level	Class	#disc.patterns	#patterns	Proportion
B	FP	6	9	66.67%
	NFP	12	12	100.00%
	OTHER	13	16	81.25%
C	Cr	3	10	30.00%
	NCr	4	11	36.36%

Table 8 displays some representative discriminating attribute values according to the two configurations and the two sets of classes. An attribute value is said to be discriminating if it does not appear in any pattern of the other classes. This experiment aims at observing the impact of texture dynamic values on the extracted patterns.

Some conclusions can be drawn. First of all, the class *OTHER* does not contain discriminating value independently of the configuration. Second, a very interesting and promising result is that the default configuration contains much more discriminating values than the NDVI configuration. Moreover, these discriminating values concern the texture attributes. This result reinforces our idea that texture attributes are very useful in automatic landscape recognition.

To conclude this experiment study, we have empirically shown that:

1. The dimensionality reduction method allows reducing the search space by deleting useless attributes.

2. Most of the extracted patterns are discriminating.
3. It appears to be more difficult to distinguish between *Cr* and *NCr* classes than *FP*, *NFP* and *OTHER* classes with our approach.
4. Most of the discriminating attribute values concern the texture attributes.

Table 8. Some discriminating dimension values per class with *minSupp=0.3* (top: default config. / bottom: NDVI config.)

Level	Class	Attribute	Value
B	FP	modis homogeneity 1km	0.48-0.52
		modis variance 1km	3.27-4.34
		modis dissimilarity 1km	1.36-1.51
		distance village	3150-6205
	NFP	modis contrast 1km	5.35-6.54
		modis dissimilarity 1km	1.51-1.66
	OTHER	NONE	
C	Cr	modis dissimilarity 1km	1.21-1.36
		modis variance 1km	3.27-4.34
	NCr	modis variance 1km	10.28-14.15

	Class	Attribute	Value
B	FP	NONE	
	NFP	rainfall	800
		distance village	3149.3-6205.6
	OTHER	NONE	
C	Cr	NONE	
	NCr	distance village	3149.3-6205.6

7 Related Work

Applications of sequential pattern mining methods to Satellite Image Time Series (SITS) include Julea et al. (2006, 2008, 2011) and Petitjean et al. (2010). Interest in these methods to study change detection on satellite images come from the fact that they are (i) multi-temporal, (ii) robust to noise, (iii) able to handle large volumes of data, and (iv) capable of capturing local evolutions without the need for prior clustering.

In Julea et al. (2008), sequential pattern mining is applied to study change in land cover over a 10 months period on a rural area of east Romania. Pattern extraction is used to group together SPOT pixels that share the same spectral evolution over time. The SITS data is thus processed at the pixel level, by taking the values of the pixels on each of the SPOT

bands. A method is proposed to visualize the extracted patterns on a single image.

Petitjean et al. (2010) present a similar approach, but pixel values are computed from four SPOT bands instead of a single band. The SITS period coverage is also much longer: a 20-year time image series is mined in order to study urban growth in the southwest of France. A visualization technique is proposed to locate areas of evolution. Results show that mining all pixels of the images leads to the generation of a huge number of non-evolution patterns. Additional strategies are then required to filter out all non informative patterns.

To the best of our knowledge, sequential pattern mining has only been applied at the pixel level on high resolution images without taking into account external data or texture information in the mining process. In this paper, we have shown that sequential pattern mining can help to characterize cultivated areas from moderate resolution remote sensing images MODIS.

8 Conclusions

The objective of this study was to propose an original method to extract sets of relevant sequential patterns from MODIS time series that can be used for cultivated area mapping.

We have developed a data mining method based on two steps and applied it in Mali. The algorithm we used was selected on the basis of its efficiency to spatio-temporal data and its scalability. An experimental study conducted on this data set reinforces our intuition about the importance of texture attributes to improve the automatic landscape recognition.

Our future work will be aimed at validating the extracted patterns per class. After which, we can go a step further to build the classifier based on these patterns and evaluate the predictions of the cultivated area at a national scale. It would be interesting to compare such a classifier with classic prediction approaches in order to evaluate the interest of data mining methods in the remote sensing domain

References

Agrawal, R., Srikant, R. (1995) Mining sequential patterns, in: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3-14.

Catlett, J. (1991) On changing continuous attributes into ordered discrete attributes, in Machine Learning EWSL-91, Springer, pp. 164-178.

Haralick, R. (1979) Statistical and structural approaches to texture image type analysis, in: Proceedings of IEEE, 67, pp. 786-804.

Haralick, R., Shanmugam, K, Dinstein, I. (1973) Textural features for image classification, IEEE Transactions on Systems, Man and Cybernetics, 3(6), pp. 610-621.

Julea, A., Méger, N, Bolon, P. (2008) On mining pixel based evolution classes in satellite image time series, in: Proceedings of the 5th Conf. on Image Information Mining: pursuing automation of geospatial intelligence for environment and security (ESA-EUSC 2008), 6 pgs.

Julea, A., Méger, N., Bolon P., Rigotti, C., Doin, M.P., Lasserre, C., Trouvé, E., Lazarescu, V. (2011) Unsupervised spatiotemporal mining of satellite image time series using grouped frequent sequential patterns, IEEE Transactions on Geoscience and Remote Sensing, 49(4), 14 pgs.

Julea, A., Méger, N., Trouvé, E. (2006) Sequential patterns extraction in multi-temporal satellite images, in : Proceedings of the 17th European Conference on Machine Learning and 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2006) - Berlin (Germany), Berlin (Allemagne), September 2006, pp. 94-97.

Petitjean, F., Gançarski, P., Masseglia, F., Forestier, G. (2010) Analysing satellite image time series by means of pattern mining, in: Proceedings of Intelligent Data Engineering and Automated Learning - IDEAL 2010, 11th International Conference, Paisley (UK), September 1-3, 2010, Springer, Lecture Notes in Computer Science, 6283, pp. 45-52.

Plantevit, M., Laurent, A., Laurent, D., Teisseire, M., Chong, Y.W. (2010) Mining Multi-Dimensional and Multi-Level Sequential Patterns, in : ACM Transactions on Knowledge Discovery from Data (TKDD), January 2010, 1), 37 pgs.

Rouse, I. (1974) The explanation of culture change, Science, 185, pp. 343-344.

Road Network and Mobility Research

A Comparison of the Street Networks of Navteq and OSM in Germany

Ina Ludwig, Angi Voss, Maike Krause-Traudes

Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS), Sankt Augustin, Germany
InaLudwig@gmx.de, {angi.voss, maike.krause-traudes}@iais.fraunhofer.de

Abstract. In Germany, the data of the Open Street Map project has become available as an alternative to proprietary road networks in commercial business geomatics software and their customers are wondering whether the quality may be sufficient. This paper describes an implemented methodology to compare OSM street data with those of Navteq for all populated roads in Germany. As a unique feature, the presented methodology is based on a matching between the street objects of OSM and Navteq and all steps are fully automated so that they can be applied to updated versions of both data sets.

While there are considerable qualitative differences between regions, towns, and street categories, at a national level the relative completeness of objects, their relative precision, and the relative completeness of names are high enough for maps. However, other attributes, which are needed for the computation of catchment areas, are still relatively incomplete.

1 Introduction

In order to compute and visualize points of sale with their catchment areas and market potentials, business geomatics applications need data of street networks. Until recently, the choice has essentially been between the products of Navteq (www.navteq.com) and Teleatlas (www.teleatlas.com). Their licensing costs may form a considerable portion of the entire costs of

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_4,

a solution, so that especially small enterprises have not been able to afford this level of analysis at all.

Since about 2009, a third alternative has entered the market: the road network data of the Open Street Map project (www.openstreetmap.org, OSM for short). It is free of charge under the Creative Commons Attribution-Share Alike 2.0 license, which is to be switched to an Open Database License. Besides its data, the OSM project itself provides web services which are relevant for business geomatics: geocoding, routing, and computation of catchment areas based on driving and drive-time distances in the road network (Neis und Zipf 2006, Neis et al. 2010, Schmitz et al. 2008).

For Germany, Infas GEOdaten (www.infas-geodaten.de) offers a data set called "OSM+", which enriches OSM data with post code areas, administrative borders, and ranges of house numbers. In its software "Filialinfo", IVU (www.ivu.de) integrated a web service for OSM data. LOGIBALL (www.logiball.de) offers a tool called "Quality Grid" to assess the completeness of the OSM road network. The enterprises explain that their customers are asking for OSM data because they are free and up to date. As another argument, in principle, everyone can extend and improve OSM data where they may be insufficient.

Given these advantages, customers of business geomatics applications are eager to know how the quality of OSM road data differs from the quality of the commercial products. If OSM road data turn out to be fit for their use, they may ask not only for OSM data, but also for market data, which are currently offered as thematic attributes only for the commercial road networks.

This article is about our first endeavor to compare the road networks of OSM and Navteq, which resulted in a diploma thesis (Ludwig 2010). Although the data structures of both road networks are global, there may be country-specific pragmatics. However, to limit the scope of our initial analysis, we restricted the comparison to Germany and to streets, i.e. populated roads, which are most relevant for site analysis in business geomatics.

Our approach differs from several others as it is based on matching road objects from the Navteq and OSM data sets and an object-wise comparison of geometries and thematic attributes. Such a match is necessary in order to compare or transfer thematic attributes between the two data sets.

As a unique feature, our methodology is highly automated. It produces a table with quality information for each single Navteq street object. For arbitrary aggregations of this table quality measures can be computed for different aspects such as completeness of objects and thematic attributes, spatial differences and differences between corresponding attributes.

The next sections describe related work, our approach, the quality measures, some results and potential future work.

2 Related work

Data quality has been defined as ”a measure of the difference between the data and the reality they represent” (Goodchild and Foreword 2006). However, our goal is not to produce any absolute measurements of quality, but to compare OSM street data to Navteq street data as a reference set. Alternatively, we could have compared Navteq data to OSM data as the reference set, but the primary question of our users is whether to switch from Navteq to OSM and not vice versa.

Quality evaluation of road data by comparison with a reference set is not a new approach. Several methods are described by (Joos 2000). An empirical comparison of OSM data to Teleatlas data in Germany is presented by (Zielstra and Zipf 2010). For OSM roads in the UK, Hakley (2009) buffered British Ordnance Survey data to determine which percentage of the OSM roads were covered. Also, he created a raster, summed up the lengths of the roads in each cell and compared them. Eng (2009) extended this work to the OS Master Map for selected parts of London. He additionally compared completeness of road names. In order to obtain a 1:1 correspondence between both data sets, he edited the data sets, adding names and merging objects in the Master Map.

We did not find any previous work which compares the modeling concepts of OSM and Navteq, or any work which attempts an automatic matching of the road objects in both data sources. However, a comparison based on matching objects is required in order to compare thematic attributes, like street names, velocity restrictions, restrictions to pedestrian etc. A matching will also be needed in order to transfer thematic market-related attributes from Navteq’s street objects to OSM’s.

Data matching is commonly used for fusing geographical data sets, e.g. data sets with different spatial resolutions or data models (Devogele et al. 1998). A commonly applied technique for matching different road networks is graph matching (Zhang and Meng 2007). However, it would have required further preprocessing to validate whether and where the OSM data set is indeed a routable graph. Suitability of OSM data for navigation is investigated in Zielstra and Zipf (2010).

Data quality distinguishes internal and external aspects, where the latter depend on the intended usage (ISO 2004). We focus on external quality, to

address the question whether OSM street data can replace Navteq street data for site analysis in business geomatics.

For internal quality the OSM project provides various validation tools. They check for instance, whether name and highway tags are assigned, whether objects are connected or self intersecting and whether any nodes are duplicates. Nevertheless, not all quality checks are available for every country. Osmose (Open Street Map Oversight Search Engine, http://osmose.openstreetmap.fr/), for example, only checks data quality for France.

In ISO 19113 (ISO 2002) five quality criteria are proposed: completeness (with errors of omission and commission), logical consistency, absolute and relative positional accuracy, temporal precision, and precision of attributes, which can be classifications, quantitative and non-quantitative attributes. Among them, we investigate completeness with errors of omission, relative positional accuracy, and relative precision of attributes. We do not investigate errors of commission, because we do not compare the full data sets and we do not measure any absolute precision. We do not investigate logical consistency, which is closely related to internal quality, or temporal precision, because we do not compare data updates.

Following Joos (2007), we define quality measures that can be applied to different aggregations of objects, especially spatial and thematic ones.

3 Methodology

Our process of matching is adopted from Devogele et al. (1998) and Walter and Fritsch (1999) and consists of the following, fully automated steps.

1. Preparation: This phase includes all preparatory activities to accommodate the different data models.
2. Investigation of correspondences between the data models: A matching between two road networks assumes corresponding concepts. Thus it must be determined how the classes of objects and their attributes can be related.
3. Matching and post-processing: Correspondences at object level are subsequently established by an algorithm.
4. Statistical evaluation.

a. Preparation

We used OSM data from April 2009. After the conversion of the OSM data file from XML to Oracle data format, we select the subset of geometries for Germany and discard any irrelevant attributes.

For Navteq data (Navteq 2009) we used the "Digital Data Streets" data product from DDS, released in July 2008. We derive new attributes for road category, foot path, pedestrian precinct, length and direction. The category indicates a road's significance on an ordinal scale: 1 = main road, 2 = 1st class road, 4 = 2nd class road, 5 = 3rd class road, 7 = forth class road). At "pedestrian precinct" cars or taxis are not permitted, while utility vehicles or ambulances may pull in, and "foot path" specifies an exclusively pedestrian usage. "Direction" is based on the original attribute delivered by Navteq slightly modified by the additional characteristic that these streets are not exclusively issued for special-purpose vehicles.

Concentrating on streets, we remove motorways and their distributor roads from both data sets (i.e. Navteq category 1 und 2, OSM motorway, trunk, and their links), as well as any Navteq road objects without names. Doing this we assume that road categories are largely correct in both data sets and that names in Navteq are rather complete. Indeed, without a category, i.e. a highway tag, a road will not be displayed on the OSM map. In Navteq road names are crucial for geocoding addresses, and road categories are crucial for finding fast routes. In total, 98.74% of all OSM road objects and 74.31% of Navteq's were used. Figure 1 compares the selected subsets of Navteq and OSM for the city of Bonn (not selected street categories are marked in bold and grey, selected ones in black). The differences indicate that the discarded classes do not exactly match in both data sets.

Fig. 1. Selected subsets (black) of Navteq (left) and OSM (right) in Bonn.

As soon as both data sets have been loaded into the databases, there are scripts to derive attributes and create the selection.

b. Comparing the data models

Navteq's data model is compatible with the standardized GDF-Format 0, while the concepts of OSM have evolved in its community (Ramm and Topf 2010). In both models distinct objects must be established when an attribute changes its value. In order to facilitate routing, Navteq's road objects additionally end at the next intersection. This difference is a major reason why there are almost twice as many street objects in Navteq as in OSM for comparable street categories in our selection.

Secondly, OSM has the highway tag to distinguish roads for cars, bicycles, pedestrians etc. Therefore, a street with several lanes has to be represented by several parallel street objects, so-called bundles. Navteq, instead, has a single object with different lanes.

Further, roundabouts, public open spaces, and their access roads can be represented differently. In Navteq, a place can be represented as one or several road objects. In OSM, a place can be an area or a road object; parking places can even be represented as a point. Access roads can be named in Navteq and need not be named in OSM. Roundabouts can be one object in OSM and several objects in Navteq. Nevertheless, compared to road intersections, the impact of these conceptual differences is low.

Table 1. Related attributes in OSM und Navteq

Navteq		***OSM***	
category	ordinal	**highway**	nominal, partly ordinal
primary name, secondary name	nominal (written with ss)	**name,** **ref**	nominal (written with ß)
direction	four nominal values	**one-way**	binary
speed limit	nine ordinal speed classes	**speed limit**	metric continuous speed values
foot path, pedestrian precinct	binary	**path, footway, steps, track, cycleway**	various nominal values of highway

Table 1. lists corresponding thematic attributes in OSM and Navteq and their scales of measurement. The value ranges of the attributes have

straightforward correspondences, with the exception of the road category. Navteq's category facilitates routing; the categories distinguish subnets of different velocity. In particular, Navteq's category 7 for fourth class roads corresponds to many highway values in OSM. Table 2 shows how we associate categories.

Table 2. Associating Navteq categories and highway values in OSM (grey indicates correspondence)

OSM \ Navteq	1	2	4	5	7
Motorway, Motorway link					
Trunk, Trunk link					
Primary, Primary link					
Secondary, Secondary link					
Tertiary					
Unclassified					
Residential					
Living street					
Service					
Footway					
Path					
Pedestrian					
Track					
Cycle way					
Steps					

c. Establishing object correspondences

A matching between two sets of objects is easier when the relation is essentially 1:1. As already explained, the selected Navteq data set is about twice as large as the selected OSM data set mainly because OSM roads need not end at the next intersection and hence are longer. Therefore we split the OSM objects into "segments" by intersecting them with buffers around the Navteq objects. Then we establish correspondences between each Navteq object and its OSM segments. This should approximate a 1:1 relationship. As a result, in average each OSM object is associated – via its segments – to two Navteq objects.

The initial set of candidates for a Navteq object shall be large, so that we do not miss any right matches. Afterwards the candidate set shall be restricted to include only the right matches. The following steps are performed to create and then reduce the candidate matching partners of each Navteq object:

1. segmentation of OSM objects with corresponding categories;
2. computation of similarity predicates;
3. matching of best ranking candidates;
4. post-processing of bad matches; and
5. evaluation of the matching.

Segmenting OSM

We create OSM segments by an intersection with buffers around each Navteq object. After some visual examinations we decided for buffers sizes of 5 m, 10 m and 30 m. 5 m is the deviation from the street center line tolerated by Navteq in Europe and the USA. As shown in Figure 2, 5 m buffers (bold grey corridor lines) may create OSM segments with multi-geometries (inner single grey line) which are rather short compared to the Navteq object. As a remedy, the 10 m buffer is introduced (outer light grey corridor). With some places, even 10 m buffers can fail, but a 30 m buffer allows the right partner to be captured as a candidate.

Fig. 2. (Multi-)geometries in buffers of 5 m and 10 m around one Navteq object

A Navteq object's initial set of candidates consists of all OSM segments contained in any of its buffers. Only OSM objects with compatible categories are eligible for this operation. As a side effect, from an OSM bundle, objects with incompatible categories will be discarded.

A single OSM object can spawn multiple segments not only due to the three buffer sizes, but also because it intersects buffers of nearby Navteq objects. However, such wrong candidates will be discarded in subsequent steps because they are too dissimilar.

Similarity predicates

To reduce the initial set of candidates, we should only use reliable attributes. Therefore, we compare Navteq objects and their candidates only by geometry (length), category, and name. Category and name were already assumed to be reliable for the initial selection during preprocessing.

An OSM segment in a buffer of 5 m or 10 m is considered similar in length to its Navteq object if their lengths differ at most by 25%. The computation takes into account that the buffers can increase the length of the OSM segments by maximum 2*5 m or 2*10 m, respectively. Lengths are not compared in 30 m buffers and when the OSM segment is shorter than the buffer size of 5 m and 10 m, respectively.

For the similarity of names, we consider that Navteq objects may have a name (i.e. primary name) or a number, like "B9" (secondary name), while OSM objects may have a name or a reference. We compare them all:

- LS1: Primary name (Navteq) with name (OSM)
- LS2: Secondary name (Navteq) with Ref (OSM)
- LS3: Secondary name (Navteq) with name (OSM)
- LS4: Primary name (Navteq) with Ref (OSM)

LS2 allows streets to be matched if none of them have an official street name. LS3 covers cases where Navteq's primary name is the name of a place and Navteq's secondary name is the name of a road at this place in OSM. LS4 covers cases where Navteq roads without a name have their street number assigned to the primary name.

Names are compared by the Levenshtein function, which returns the difference of letters in two strings ignoring upper and lower cases, blanks and hyphens. For a pair to be similar in names, the Levenshtein distance must return 0 for the comparison of street numbers in LS2 and LS4, and may be at most 4 for LS1 and LS3. LS1 and LS3 compare names, and 4 different characters allow a road name to contain two "ß". One "ß" already occurs in the German word for street, which is spelled "Straße" in OSM and "Strasse" in Navteq. A better approach might be to standardize the names before comparing them.

Matching by best ranking candidates

Table 3 presents the possible combinations of similarity predicates and assigns them a rank (5, 10, 30 is the size of the buffer, L means similar length, N means similar names, null means OSM candidate without a name). For example, the top rank of 1 is assigned to OSM candidates that lie in the 5m buffer and are similar in length and name. If the names are similar, but lengths are only similar in the 10m buffer, the rank is 2. In

general, similar lengths are more important than similar names because there can be OSM streets without names.

Since the rank takes several similarity predicates into account and since, the categories must not fit exactly, the matching tolerates deviations in position and form between the partners, in their category, and in the notation of their name.

Table 3. Rank definition by similarity predicates

Rank	5 LN	5 N	10 LN	10 N	30 N	5Lnull	10Lnull	5 L	10 L
1	■		■		■				
2		■	■		■				
3			■		■				
4						■	■		
5							■		
6		■		■	■				
7				■	■				
8					■				
9								■	■
10									■

In short, ranks are defined as follows:

- Rank 1-3: here names and lengths are similar in a 5 m or 10 m buffer;
- Rank 4-5: here lengths are similar in a 5 m or 10 m buffer and the OSM partner has no name;
- Rank 6-8: only names are similar;
- Rank 9-10: names are not similar, but lengths are.

Only the OSM segments with the highest rank are kept as candidates, which can be one or more. They can represent one or more OSM objects. These OSM objects are the matches of the Navteq object. The rank of the match is the common rank of the final candidates.

Post-processing

In the post-processing phase a GIS was used to visualize the ranks of the matches and to check some samples for correctness. Quite often, there were mismatches of access roads in Navteq which do not exist in OSM, see Figure 3 (Navteq: black, OSM: grey). They are falsely matched with their access road because they have the same name, although the lengths are different. Therefore we decided to remove all Navteq objects of rank 6 or 7 which have one dead end and are connected to one or more segments

with identical name at the other end. In total 25,000 such objects with their matches were removed.

Fig. 3. Access roads in Navteq (black) missing in OSM (grey)

d. Evaluating the matching

In many approaches, the quality of a matching algorithm is determined in terms of mismatches by a comparison with a manual assignment. Instead, we consider the similarity ranking where bad matches should be revealed by a bad ranking of the OSM candidate. In Figure 4 the Navteq objects are colored according to the rank of OSM matches. The darkest Navteq segments indicate a matching with OSM segments similar in name and length, followed by segments with unnamed OSM partners, lighter segments have a partner with similar name but differing lengths and the lightest ones got a partner with a strongly differing name.

Actually, many matches are correct although their rank is bad. This is the case when the OSM object has no name or is not close enough. Navteq objects with unnamed OSM partners often fall into category 7 and are paths in forests or fields.

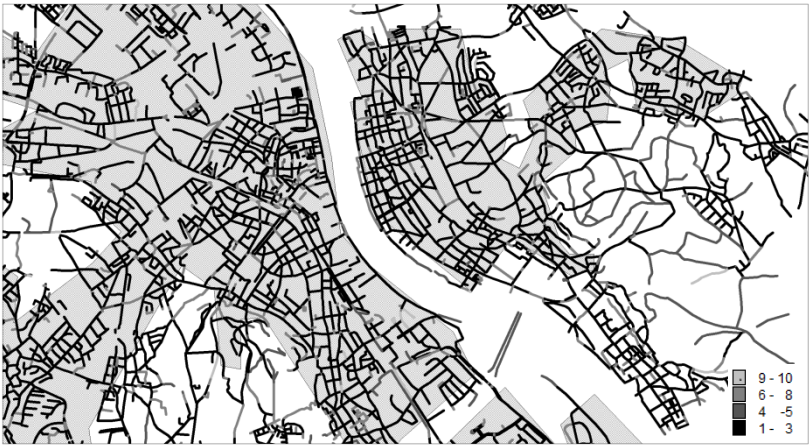


Fig. 4. Navteq roads in Bonn and the rank of their matching

In total, more than 60% of the matches have similar names and lengths in the 10 m buffer and only 3% are in the worst matching classes 9 and 10 (cf. Figure 5).

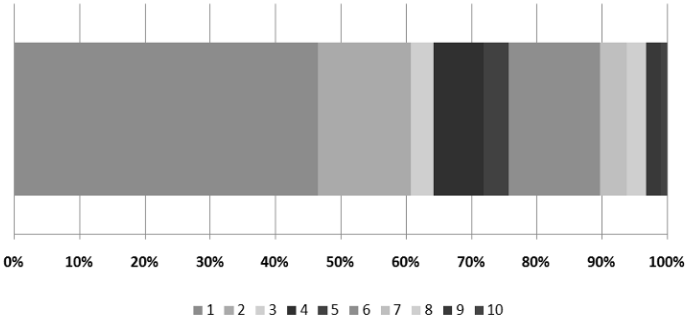


Fig. 5. Percentages of matching ranks

4 Quality assessment

We assess the quality of the OSM street data relative to the Navteq data for the criteria of relative completeness of objects and relative positional accuracy of objects. For the attributes in Table 1. we measure relative completeness, i.e. we ask if they are provided (or omitted) for both the Navteq object and its matching OSM object. For speed limit, we additionally com-

pute the deviation. All quality measures are implemented as scripts in an Oracle database.

a. Relative quality measures

For each quality criterion, we propose a quality measure that can be applied to an arbitrary set of street objects, like all streets (of a particular category) in a particular community or district.

- Relative object completeness: Given a selection of Navteq objects, determine the percentage that has a match.
- Relative attribute completeness for an attribute A ε {"name", "one-way", "pedestrian path" or "pedestrian zone", "speed limit"}: Given a selection of Navteq objects, first determine the subset that has any match and the attribute A. In this subset, determine the percentage of objects with at least one OSM match which has a corresponding attribute (correspondences according to Table 1.).

Navteq uses eleven speed limit classes, while the speed in OSM can be any number. Therefore, to compare deviations in speed limit, we first discretize the range: 1: < 30 km/h, 2: <50 km/h, 3: < 70 km/h, 4: < 100 km/h, 5: ≥ 100 km/h. Then for the speed classes 2 - 5 we check whether matching partners do not have speed limits in the same interval:

- Difference in speed limits: Given a selection of Navteq objects, first determine the subset that has any match and where the speed limit is > 30 km/h. In this subset, determine the percentage of objects with at least one OSM match which either has no speed limit or a speed limit in a different interval.

We use buffer sizes to measure positional differences. Since we do not know the exact metric differences, we only count the percentage of objects whose matches lie in the 5 m, 10 m, or 30 m buffer.

- Positional difference within 5 m: Given a set of OSM objects, determine the percentage that has a match and all its matches are of rank 1, 2, 4, 6, or 9. According to Table 3. this means the segment lies only in 5 m buffers.
- Positional difference beyond 10 m: Given a set of OSM objects, determine the percentage that has any match and at least one of rank 8. This means the segment lies in at least one 30 m buffer.
- Positional difference 5 m – 10 m: Given a set of OSM objects, determine the percentage that has any match and at least one is of rank 3, 5, 7, but none is of rank 8. This means the segment lies in at least one 10 m buffer, but not in a 30 m buffer.

b. Some results

Positional difference

In total, 73% of the OSM street objects in Germany are within a distance of 5 m, 21% between 5 m and 10 m, and 6% between 10 m and 30 m away from their Navteq partner. Positional differences increase as the Navteq category increases, as shown in Figure 6. We detected higher deviations in parks and in the countryside, where there may be less visual cues for correcting a track than in a town. Alternatively, the precision of Navteq might be lower in such areas. Higher deviations may also be caused by OSM trackers walking on the sideway, and by places that are represented by different kinds of objects in OSM and Navteq, as described in section 3.2.

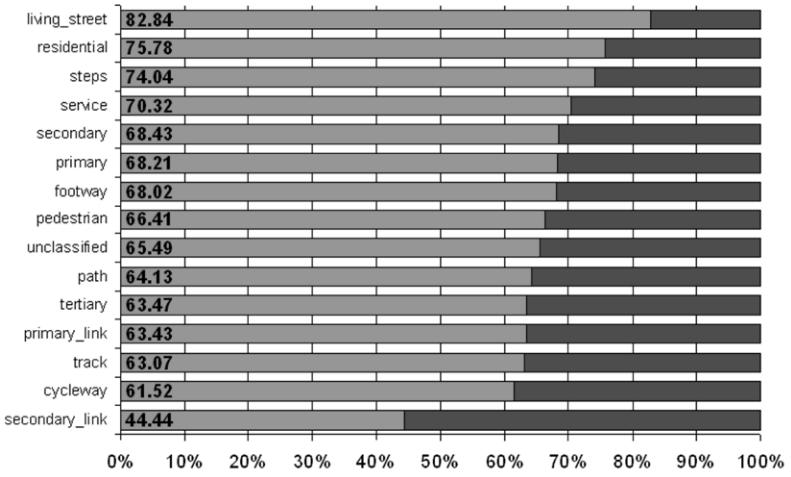


Fig. 6. Positional differences of OSM objects to matching Navteq objects per Navteq street category (light grey = 5 m, dark grey = 10 or 30 m)

Relative completeness of attributes

In general, the percentage of missing names increases from inhabited areas (5.6 %) to uninhabited ones (17.5%) and from important to less important streets (13.8% in Navteq category 7, 13.8% in category 5, and 4.7% in category 4). It is again paths in parks and in the countryside where OSM partners are lacking of a name. Names are also missing with OSM objects representing places or accessing car parks.

The attribute "oneway" in OSM as compared to Navteq is also more often missing in uninhabited areas (48.8%) than in inhabited ones (28.1%). Regional differences can be considerable. For instance, in the towns of Bottrop, Hamm, and Heilbronn, more than 40% of the Navteq objects with this attribute also have an OSM partner with this attribute, but in Osnabrück it is less than 20%. The reason may be that for Osnabrück data were imported to OSM from the FRIDA project (http://frida.intevation.de/ueber-frida.html), but subsequently not checked or updated.

For the Navteq attribute "footpath," 53% have a partner with a corresponding attribute and for the Navteq attribute "pedestrian precinct," it is 64%. In contrast, for the attribute "speed limit," the relative completeness of OSM compared to Navteq is dramatically low. It is missing for 80.7 % of the objects in inhabited areas and for 92.6 % of the objects in uninhabited areas.

The differences in relative attribute completeness clearly confirm that the focus of the OSM community has been somewhat complementary to that of Navteq, the latter focusing on motorized and the former on non-motorized use.

Difference in speed limits

The differences in speed limits can be analyzed in a confusion matrix. Table 4. shows how often corresponding Navteq and OSM objects fall into different speed limit intervals. In speed class 2 (30 km/h – 50 km/h, column 2) the coincidences are highest. Row 6 is most dramatic, which contains all cases where a speed attribute is missing in OSM.

Table 4. Confusion matrix for speed limit intervals in Germany (in percentages)

OSM \ *Navteq*	*1*	*2*	*3*	*4*	*5*
1	2.74	0.28	0.09	0.01	0.00
2	6.37	18.01	3.36	0.47	0.08
3	0.28	0.82	6.72	1.90	0.97
4	0.01	0.01	0.26	2.35	1.23
5	0.00	0.01	0.24	1.57	2.70
No speed limit	90.60	80.86	89.34	93.7	95.02

Speed limit is important for computing the driving times of routes in Navteq. It is complete and the precision should be high. Due to the relative incompleteness in OSM the open route service uses the street category (the highway tag) to determine drive times, e.g. 110 km/h for primary roads, 40km/h for residential ones.

Completeness of objects

In general, relative object completeness decreases from 79.8% in inhabited areas to 50.8% in uninhabited ones. It also decreased from 92.4% for Navteq category 4 streets, via 88.3% for category 5 streets to 54.6% for category 7 streets. Fig. 7 displays the completeness of objects aggregated to German districts and Figure 8 for its 15 largest cities.

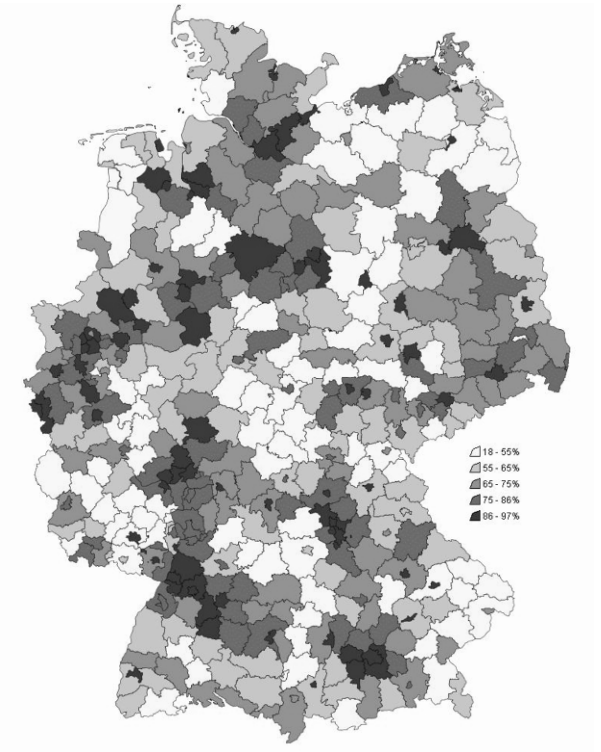


Fig. 7. Relative completeness of objects per district in Germany (class borders according to quartiles)

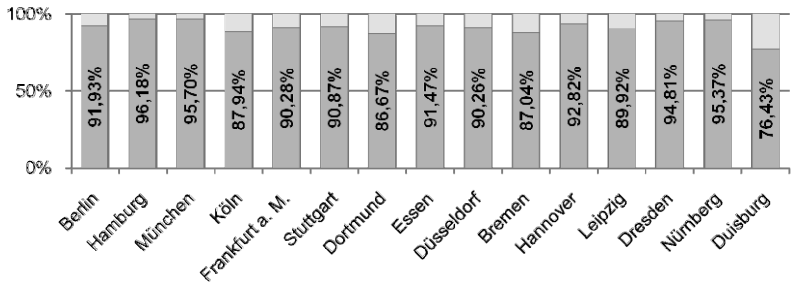


Fig. 8. Relative completeness of objects in cities (percentage of associated Navteq objects)

Completeness of streets indeed varies regionally; it ranges between 97% in densely populated areas to 18% in uninhabited regions. Also it decreases from 92% for highly frequented streets (Navteq category 4), via 88% for Navteq category 5 streets, to 55% for Navteq category 7 streets.

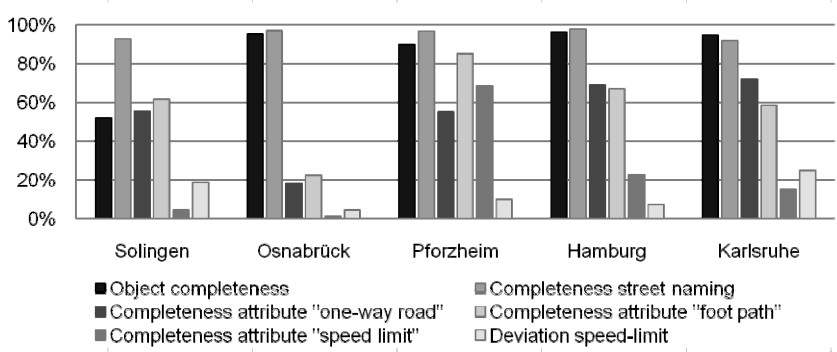


Fig. 9. Examples of OSM object completeness and thematic attribute completeness and correctness

Figure 9 shows that there are towns with a relatively high completeness of streets, but a relatively low completeness of thematic attributes and vice versa:

- Solingen: only 46% completeness of road objects – but 93% of them without names.
- Osnabrück: 90% completeness of road objects – but thematic attributes only for 18% of the one-way roads, 22% of the pedestrian roads, and 1.5% for speed limits.
- Pforzheim: 68% of the roads have speed limits and only 10.3% of the limits deviating from Navteq's – but only 55% of the one-way roads are identified.

- Hamburg: all analyzed attributes are very complete – except for one-way roads (only 69%).
- Karlsruhe: 90% completeness of road objects and nearly all of them have names – but 85% have no speed limits and 25% of them deviate from Navteq's.

5 Conclusions

In this article we compared the data models of the road networks of OSM and Navteq and pointed out three differences. In particular, OSM road objects may be longer, so that there are almost twice as many objects in the Navteq data set. Therefore we segmented OSM objects by drawing buffers around Navteq objects, before we established correspondences between these two sets. The matching is based on a rank, which considers similarity in lengths, names, and categories. For arbitrary aggregations of objects, we defined measures for relative object completeness, relative attribute completeness, and relative positional and relative thematic accuracy. All measures count the portion of objects whose matches satisfy a suitable condition.

At the national level, the quality of OSM is highest regarding relative object completeness. The relative completeness of attributes seems to be the higher the more relevant the attribute is for non-motorized usage. Quality differs locally, and even in a single town the different aspects of quality may vary. Therefore it is important that quality measures can be applied to arbitrary selections of objects.

For a business geomatics application, the user should look at the OSM quality in the target area. If it reaches the national average regarding relative object completeness and relative precision, OSM maps can be used for display. For computing catchment areas around a facility, the quality of the relevant street categories should be considered: depending on the type of facility, higher or lower street categories may be relevant. Since the speed limit attribute in OSM is quite incomplete compared to Navteq, the open route service relies on the highway tag for estimating drive times. Catchment areas based on drive time (3 – 10 – 15 minute drive time zones) should be computed using the highway tag rather than the speed limit of OSM objects. Even catchment areas based on driving distance (km) are not reliable if the relative completeness of the attributes paths, footways, steps, tracks, cycleways, and one-ways in the area of interest is low.

The OSM data set is continuously growing and improving (http://wiki.openstreetmap.org/wiki/Stats). Already during our investiga-

tion in 2009 some OSM dead end roads in the city of Heidelberg were corrected in the sense of Navteq. Thus, it is important to repeat our assessment. This can be done quite easily since the methodology is highly automated. We plan to repeat the analysis for new Navteq data sets for 2010 and some following years.

We want to extend our approach to all roads in Germany, and to more countries in Europe. We could also switch the focus and assess Navteq's data relative to OSM's. Last but not least, the available matching can be used to transfer thematic market data from Navteq road objects to OSM objects and use OSM to estimate market potentials in business geographics.

References

Devogele, T., Parent, C and Spaccapietra, S. (1998) On spatial database integration. INT J GEOGR INF SCI, 12 (3), 335 – 352.

Eng, M. (2009) A quality analysis of OpenStreetMap data. Dissertation. University College London.

Goodchild, M.F. (2006) Foreword, In: Devillers, R and Jeansoulin, R (ed.) Fundamentals of Spatial Data Quality. ISTE, London, 13–16.

Hakley, M. (2009) How good is OpenStreetMap information? A comparative study of OpenStreetMap and Ordnance Survey datasets for London and the rest of England. Retrieved December, 10, from London's Global University: http://www.ucl.ac.uk/~ucfamha/OSM%20data%20analysis%20070808_web.pdf

ISO 14825 (2004) Intelligent transport systems - Geographic Data Files (GDF) - Overall data specification. City.

ISO 19113 (2002) Geographic information - quality principles. International Organization for Standardization. Genf.

Joos, G. (2000) Zur Qualität von objektstrukturierten Geodaten. München, Universität der Bundeswehr München, Dissertation.

Ludwig, I. (2010) Abbildung von Straßendaten für Qualitätsuntersuchungen – Ein Vergleich von OpenStreetMap mit Navteq, Universität Bonn, Diploma Thesis.

Navteq (2009) Navteq's NAVSTREETS Street Data: Reference Manual v3.0. 2008. – Retrieved December, 10, 2009, from: http://faculty.unlv.edu/ jensen/ gisdata/navteq/TechnicalReference/NAVSTREETS_Reference_Manual_v3-.pdf.

Neis, P., Zipf, A., Helsper, R., Kehl, A. (2006): Webbasierte Erreichbarkeitsanalyse – Vorschläge zur Definition eines Accessibility Analysis Service (AAS) auf Basis des OpenLS Route Service. REAL CORP. Wien. Retrieved June, 22, 2010, from: http://tolu.giub.uni-bonn.de/karto/CORP07-AAS-pn-az-final.pdf

Neis, P., Zielstra, D., Zipf, A., Struck , A. (2010) Empirische Untersuchungen zur Datenqualität von OpenStreetMap – Erfahrungen aus zwei Jahren Betrieb mehrerer OSM-Online-Dienste. AGIT 2010. Symposium für Angewandte Geoinformatik. Salzburg. Austria

Frederik Ramm, Jochen Topf, Steve Chilton (2010) OpenStreetMap: Using, and Contributing to, the Free World Map , UIT Cambridge.

Schmitz, S., Neis, P., Zipf, A. (2008) New applications based on collaborative geodata – the case of routing. Submitted for: XXVIII INCA International Congress on Collaborative Mapping and SpaceTechnology, Gandhinagar, Gujarat, India. Retrieved June, 22, 2010, from: http://tolu.giub.uni-bonn.de/karto/publications/pdf/conference/cmap2008.cartography-bonn.subm.pdf

Walter, V. and Fritsch, D. (1999) Matching spatial data sets: a statistical approach. INT J GEOGR INF SCI, 13 (5). 445 – 473.

Zhang, M. and Meng, L. (2007) An iterative road-matching approach for the integration of postal data. Computer, Environment and Urban Systems, 31,597 – 615.

Zielstra, D. and Zipf, A. (2010) A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany. AGILE 2010. The 13th AGILE International Conference on Geographic Information Science. Guimarães, Portugal.

Extension of Spatial Correlation Analysis to Road Network Spaces

Toshihiro Osaragi, Tomoyuki Naito

Department of Mechanical and Environmental Informatics,
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, Japan
osaragi@mei.titech.ac.jp, naito@os.mei.titech.ac.jp

Abstract. The spatial correlation analysis is proposed to analyze urban activities quantitatively. This paper describes an extension of spatial correlation analysis defined in a two-dimensional Euclidean space to a road-network space. We discuss a method for applying spatial correlation analysis to a road-network space and illustrate the details of computation methods. By using actual GIS data as numerical examples, a comparison of the results from the Euclidean distance and the network distance is shown. Also, we demonstrate some case studies using a variety of computation methods.

1 Introduction

Assessing the current state of a city or region is essential to the process of urban and regional planning. Some of the methods proposed for quantitatively assessing various spatially distributed characteristics include the K function method (Ripley 1981), nearest neighbor method (Clark and Evans 1954, 1955), join analysis (Berry and Marble 1968), clump analysis (Roach 1968), Moran's I statistics, and spatial correlation analysis (Cliff and Ord 1973). The spatial correlation analysis methods investigated in this study utilize a spatial correlation function that extends the concept of the correlation function to a two-dimensional space to analyze spatial interrelationships among city activity elements at two distance-separated

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_5,

points. The origin of the spatial correlation analysis method dates back to the "quantitative revolution" that occurred within the discipline of geography between the late 1950s and early 1960s, largely in the United States. At the core of the discussion at that time was the concept of spatial autocorrelation, which describes the spatial interrelationship between elements of the same type. This concept, developed and generalized by Cliff and Ord (1973), was applied not only in geography, but also in a wide range of other fields such as econometrics, ecology, epidemiology, and urban planning. This concept also served as the basis for spatial analysis methods such as the K function method and join analysis (Getis 2008).

The achievements of Cliff and Ord were reappraised in recent years (Griffith 2009). The spatial autocorrelation, which was considered a problem in standard inferential statistics, is now considered a fortunate characteristic of a wide range of spatially distributed phenomena (Goodchild 2009). Namely, spatial autocorrelation is a description of relationship between the degree of similarity between observations and the distance separating them (Fotheringham 2009). Spatial correlation analysis methods are now used to assess the spatial relationships between urban activity elements in the fields of urban planning and architecture. In Japan, beginning with a series of studies by Aoki (1986a, 1986b), these methods were extended to techniques for quantitatively analyzing the spatial influence relationships between urban activity elements of different types (Aoki 1987).

In a series of analyses relating to spatial correlation analysis methods, the distances between urban activity elements have been defined in terms of Euclidean distance under the assumption of a two-dimensional continuous space. In real urban spaces, however, people and physical objects do not move along straight lines, but rather along roads; therefore, in analyzing the spatial relationships between facilities, the conventional assumptions that have been made are considered to be overly simplistic. Koshizuka and Kobayashi (1983) suggested that within urban areas of high road density there is considered to be a relationship of noticeable proportions between these two separations. But, if an attempt is made for a more detailed analysis of urban activity elements distributed along a network, it is more desirable to use network distance in a network space than to use Euclidean distance within a continuous space.

In view of this requirement and increasing precision of digital spatial data made possible in recent years, analysis methods that can be applied to various kinds of facilities on a road network have been developed, most notably by Okabe et al. (Yomono 1993; Kitamura and Okabe 1995; Yamada and Okabe 2000; Okunuki et. al. 2005; Okabe et al. 2009).

Many researches on network autocorrelation have so far been done (Black 1992, Black and Thomas 1998, Leenders 2002, Peeters and Thomas

2009). However, many of them have regarded spatial autocorrelation as a problem which should be solved in a spatial regression model. Moreover, spatial relationships of locations were described using weight matrix and the values of spatial autocorrelation changing according to the size of spatial lag have not been discussed.

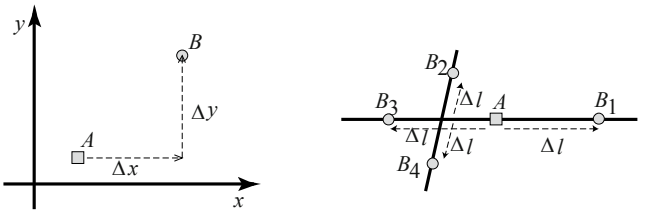
Thus, in the present study, we set out to develop methods for performing spatial correlation analysis on road networks and then to assess the effectiveness of these methods. First, as outlined in Section 2, we examine a method for applying the basic concept of spatial correlation to a network. In Section 3, we propose variations of the calculation methods; the variations are adapted to the particular objects and aims of the analysis. In Sections 4 and 5, we seek to verify the effectiveness of the proposed analysis methods by using example analyses. Finally, in Section 6, we summarize the findings of the study.

2 Network spatial correlation analysis

2.1 Basic concept

In implementing a spatial correlation analysis method, we start by defining a correlative concept for two spatially separated locations in a network. Using the spatial correlation concept described here, we can quantitatively assess the spatial continuity and interrelationship of various activities within an urban area.

We first express the network in question as a whole as L_T and its total length as l_T. The distances within this network are defined as the shortest routes between two points. At a point p on this network, the variable X_i (i=1,…, m) is taken to have a value of x_{ip}. Thus, the spatial analysis method analyzes the correlation between the variable value at a particular location and the variable value at a location that is at some specified distance away. In two-dimensional continuous space, possessing the information of "*distance*" and "*direction*" enables us to uniquely define the location of arbitrary points in the x and y directions from a reference point as Δx and Δy, respectively. However, if we consider a location Δl that is defined simply as being a certain distance from a reference point (i.e., "*direction*" information is unknown), then this could be any one of multiple points, making it difficult to establish a one-to-one correspondence between two points (Figure 1).



Two-dimensional continuous space Network space

Fig. 1. Difference between Euclidian distance and network distance

Thus, we express the set of points at a distance τ from an arbitrary point p on a network L_T as P_τ={ $p_\tau(1)$, ..., $p_\tau(n_{p\tau})$} and define the average of X_i values at these $n_{p\tau}$ points as $x_{ip}(\tau)$.

$$x_{ip}(\tau) = \sum_{k=1}^{n_{p\tau}} x_{ip_{\tau(k)}} \Big/ n_{p\tau} \,. \tag{1}$$

In the case of τ =0, $x_{ip}(\tau)$ has the value of x_{ip}. Expressing the expected value of $x_{ip}(\tau)$ for the entire network as $\overline{x}_i(\tau)$, the spatial covariance function $C_{ij}(\tau)$ can be obtained by calculating the expected values of the covariance of $x_{ip}(0)$ and $x_{jp}(\tau)$, as point p is moved over the network L_T. This is shown by Eq. (2). (Refer to Griffith (2009) in the matter of the detailed definition and mathematical formulation of spatial correlation.)

$$C_{ij}(\tau) = \frac{1}{l_T} \int_{p \in L_T} \Big(x_{ip}(0) - \overline{x}_i(0)\Big)\Big(x_{jp}(\tau) - \overline{x}_j(\tau)\Big)\, dp \,. \tag{2}$$

Using this equation we can then define the *spatial cross-correlation function* $R_{ij}(\tau)$ according to the equation below.

$$R_{ij}(\tau) = C_{ij}(\tau) \Big/ \sqrt{V_i(0) V_j(\tau)} \,, \tag{3}$$

where

$$V_i(0) = \frac{1}{l_T} \int_{p \in L_T} \Big(x_{ip}(0) - \overline{x}_i(0)\Big)^2 dp \,,$$

$$V_j(\tau) = \frac{1}{l_T} \int_{p \in L_T} \Big(x_{jp}(\tau) - \overline{x}_j(\tau)\Big)^2 dp \,.$$

$R_{ij}(\tau)$ can take values between –1.0 and 1.0. When the value of the correlation function between variable X_i at a reference location and variable X_j at a distance τ away is close to 1.0, there is "*colocalization*" between the points; conversely, if the value is close to –1.0, there is a relationship of "*exclusion*." In addition, when $i=j$, the function is determined for the case where the above definition holds true for identical variables, and this is referred to as a *spatial autocorrelation function*. When the value of this function is close to 1.0, it indicates that the spatial distribution of identical urban activities tends to be continuous, which corresponds to "*conurbation*."

2.2 Computation methods using digital map data

The above discussion assumes continuously distributed spatial variables. Next, we describe an analysis method that is built on this basic concept and makes use of actual digital map data. In conducting an analysis with actual digital map data, residential facilities and commercial buildings are handled as point-distributed variables in majority of cases. In conventional two-dimensional spatial analysis, when actual map data is employed, raster data (obtained by dividing the analyzed area into a lattice of cells of uniform shape and size) is used, and the correlation analysis is conducted by applying a density conversion to transform a point distribution to a discrete distribution. A method has also been devised for network spaces, in which the whole network is divided into cells such that each length of the network is made equal in the same way that a two-dimensional space is divided into a cell (Shiode and Okabe 2004). However, the equidistance between two separate cells is not maintained precisely, as it is in the case of raster data. That is, attempting analysis by preparing network data by dividing the network into equal lengths in advance is difficult. Thus, here we apply a method that uses randomly generated points in a network, as shown in Figure 2, where the spatial variables for the area in question are transformed into a discrete distribution based on line-buffers at regular intervals centered on the random points. Although this random-point-approach is never the only solution, the authors think this approach is easy and efficient for actual computations.

Figure 2 illustrates the computation method in detail and shows the expressions that assume a discrete distribution. We refer to Figure 2 to explain an example in which the spatial variables are location and building usage.

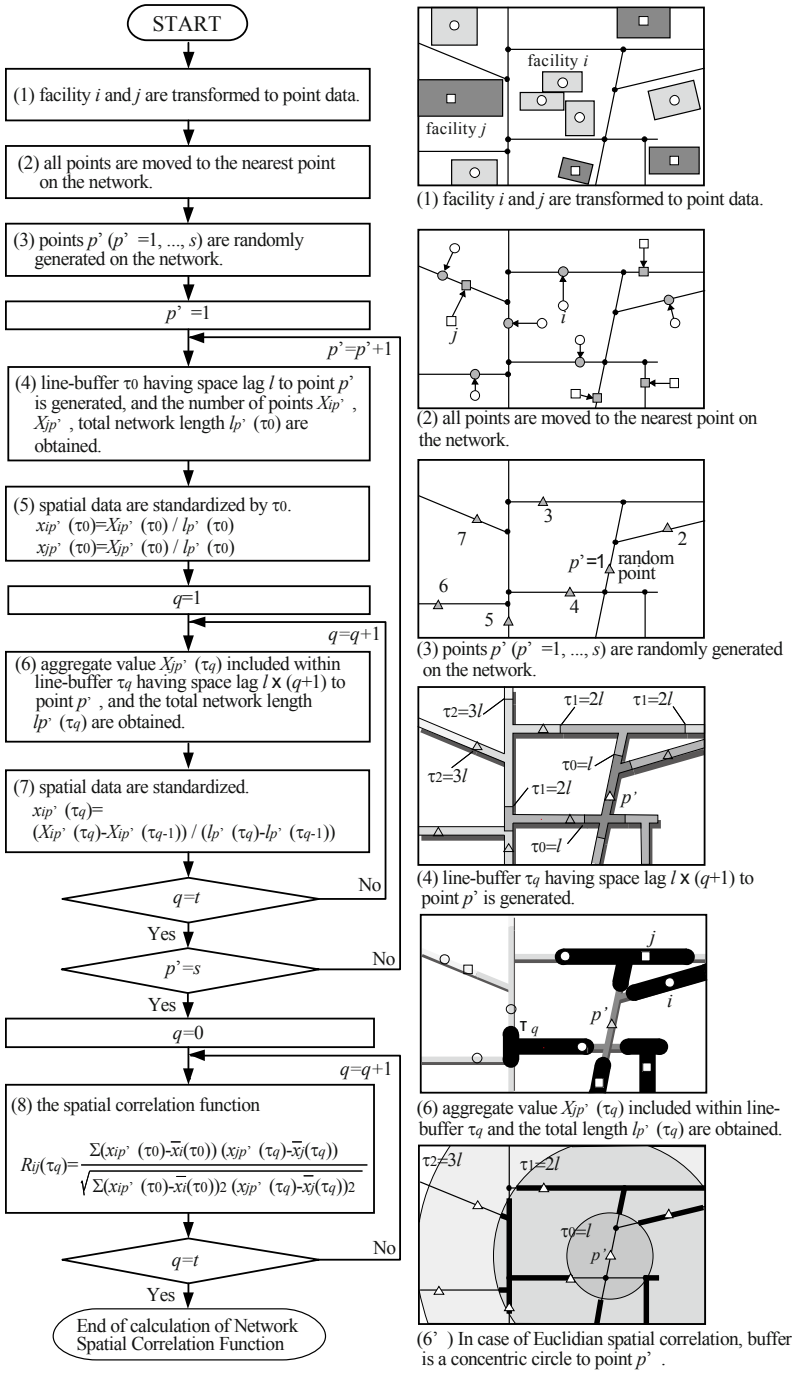


Fig. 2. Calculation method of network spatial correlation function

First, (1) the building data for the subject of analysis, facility i, is transformed to point data, and then (2) all points are moved to the nearest point on the network. Next, (3) points p' (p'=1,..., s) are randomly generated on the network. (4)(6) Line-buffers having space lag τ_q (q=0,..., t) relative to each point p' are sequentially generated, and we determine the aggregate value $X_{ip'}(\tau_q)$ included within each line-buffer relative to random point p', and total network length $l_{p'}(\tau_q)$. Then, (5)(7) we divide the value of $X_{ip'}(\tau_q)$-$X_{ip'}(\tau_{q-1})$ by $l_{p'}(\tau_q)$-$l_{p'}(\tau_{q-1})$ to determine $X_{ip'}(\tau_q)$ per unit of network length. (8) Now, we can determine the value of the *spatial autocorrelation function* $R_{ii}(\tau_q)$ as the correlation function for the spatial variable $x_{ip'}(\tau_0)$ and $x_{ip'}(\tau_q)$ obtainable for each random point p'. Also, for the subject facility j, if we similarly find the variable $x_{jp'}(\tau_q)$ and determine the correlation function relative to $x_{ip'}(\tau_0)$, we obtain the *spatial cross-correlation function* $R_{ij}(\tau_q)$.

3 Spatial variable and variations of calculation methods

The above method generates line-buffers based on spatial distances in the network, aggregates the number of facilities, and standardizes the aggregated spatial variables based on the lengths of the network within line-buffers, but it is also possible to use the calculation technique described below in accordance with the objects and aim of the analysis.

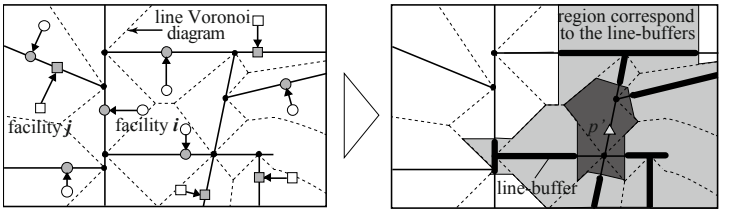
When aggregating spatial variables of facilities included in each line-buffer, apart from simply aggregating the number of facilities, it is also possible, for instance, to sum the continuous quantity of the floor area of the subject facility. Utilizing a continuous quantity is a good choice when conducting an analysis up to and including the scale of the facilities.

With spatial variables that are aggregated for each line-buffer, the total length of line-buffer for each aggregation point p' varies from point to point, making it necessary to standardize in terms of the size of the regions that correspond to the line-buffers. In addition to standardizing according to network length of line-buffer, it is also possible to standardize according to the area of regions corresponding to the line-buffers (Figure 3). The regions corresponding to the line-buffers can be obtained by dividing the line Voronoi diagram by the edges of the line-buffers. The detailed calculation-methods of Voronoi diagram based on line-objects are described in Okabe et al. (1992).

When standardizing based on the network lengths corresponding to the line-buffers, even in the case of identical facility distributions, the value of the standardized spatial variables will differ if the road density differs

(Figure 4). At the same time, if standardization is done in terms of area, the value of the spatial variables will not vary, regardless of the road density. In other words, when conducting a comparative analysis between areas having substantially different road densities, it is best to standardize by area. When analyzing housing distribution, it is necessary to use habitable land area by subtracting the area of fields, parks, etc. Note, however, average socio-economic values defined originally for areas are not necessary to be standardized, since such data are not influenced by the density of roads.

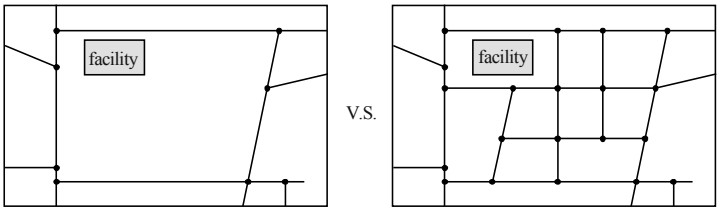
When aggregating spatial variables, it is possible to generate line-buffers based on spatial distance or temporal distance. In the spatial distance case, line-buffers are sequentially generated at fixed distance intervals, based on the network distance on the roads. Whereas, in the temporal distance case, the line-buffers are generated based on the movement distance within a fixed interval of time. The temporal distance method is preferable when the use of automobiles is assumed and when time intervals are selected to be the basis of the analysis.



A line Voronoi diagram generated by the set consisting of road links (Okabe et al., pp.170-183, 1992).

The regions corresponding to the line-buffers can be created by dividing the line Voronoi diagram by edges of line-buffers.

Fig. 3. Standardization according to the area corresponding to the line-buffers



The value standardized by road length is large in low road density.

The value standardized road length is small in high road density.

Fig. 4. Difference of standardized values according to the road density

4 Comparison of Euclidean distance and network distance

We conducted a comparison of a spatial correlation analysis based on Euclidean distance and a spatial correlation analysis based on network distance. Taking a district centered on Shimokitazawa railway station in Tokyo (Figure 5) as the subject of analysis, we determined the spatial cross-correlation function (Figure 6(1)) between office buildings and commercial facilities. When using Euclidean distance, the value of cross-correlation in the small range of the space lag is high, but at a distance of approximately 200 m, the value suddenly decreases. In contrast, when using network distance, the decrease in distance is small. Such a difference is generated because in a network space it is necessary to take detours to get access between facilities that are adjacent in a two-dimensional continuous space.

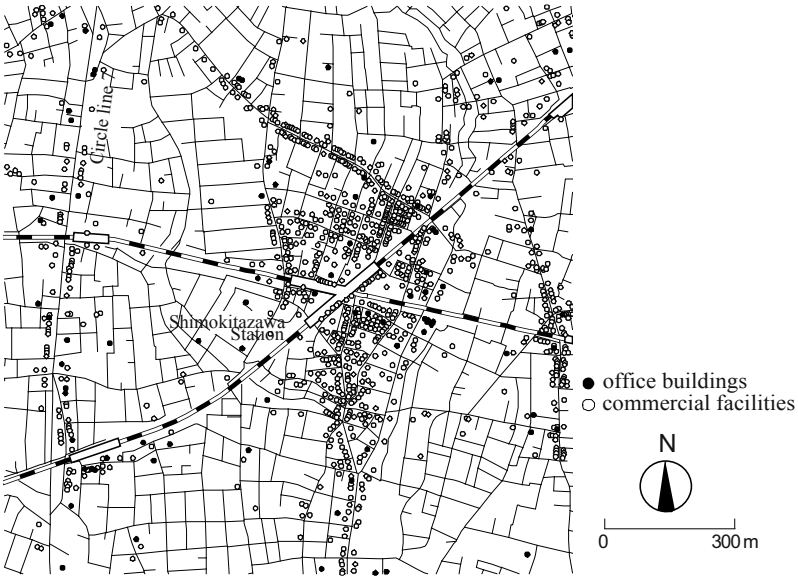
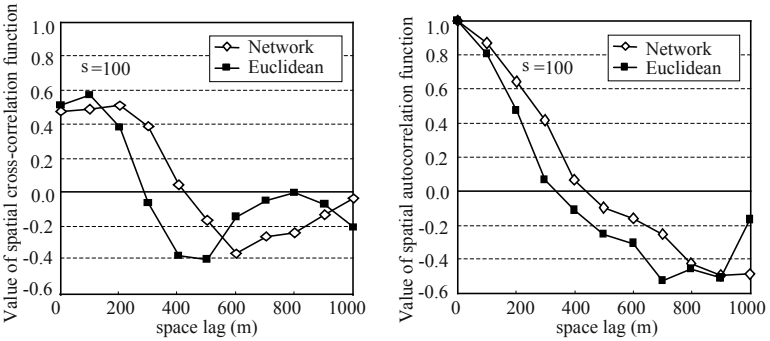


Fig. 5. Study area (Shimokitazawa in Tokyo)

Next, we determined the spatial autocorrelation function of commercial facilities (Figure 6(2)). The results show that in the case of using Euclidean distance, the distance attenuation of values is high, whereas in the case of using network distance, it is mild. This area is divided broadly by two railway lines, and the commercial facilities located in and around the train station are influenced significantly by this division. It is this factor that ac-

counts for the difference between the two cases. That is, from the viewpoint of Euclidean distance, the facilities are located adjacent to one another and form a single compact commercial district. However, in terms of actual network distance, the facilities are located a certain distance apart—between the train lines or separated by a roundabout route—so that they form a spatially diffused commercial district.



(1) Spatial cross-correlation function of office buildings and commercial facilities

(2) Spatial autocorrelation function of commercial facilities

Fig. 6. Spatial correlation function based on Euclidian distance and network distance

5 Spatial variables and comparison of calculation methods

Taking the Shimokitazawa district (Figure 5) as the subject of analysis, we determined the spatial autocorrelation function for different methods of facility aggregation (Figure 7). Looking at residential facilities (Figure 7(1)), we find that counting the number of facilities yields a value of spatial autocorrelation that is higher than that when aggregating the floor area, up to a spatial lag of 200 m. In contrast, with office buildings (Figure 7(2)), when the spatial lag is within a low range, aggregating the floor area leads to a higher value. Since the residential facilities are a complex mix that include large-scale apartments on major arterial roads and concentrations of detached houses on narrow back streets, analysis based on aggregation of the floor area results in a lower degree of conurbation. In contrast, since office buildings tend to be located continuously and on a similar scale, analysis based on floor area aggregation is observed to produce a relatively high degree of conurbation.

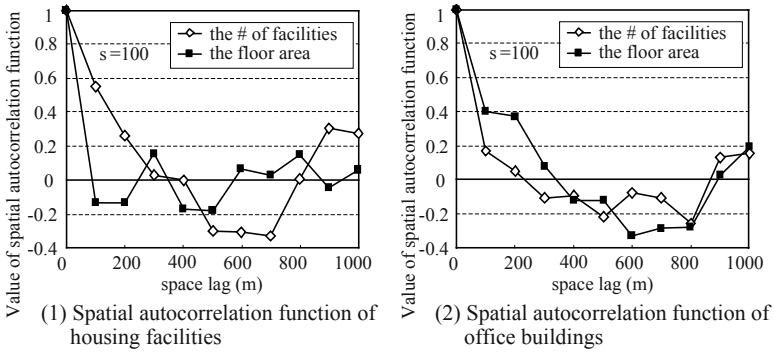


Fig. 7. Spatial autocorrelation function based on different facility aggregation

Next, we verify that the influence due to road density is smaller in the case of standardization by area than in the case of standardization by link length. Taking the Shimokitazawa district (Figure 5) again as the subject of analysis, we used the normal road network data (Figure 8(1)) and also data simplified (Figure 8(2)) by deleting the links that do not connect road ends to other roads to determine the spatial autocorrelation function for residential facilities and applied the two standardization methods to each data set. The results when standardizing by link length (Figure 8(3)) reveals a substantial difference between the normal road data and the simplified road data. In contrast, the results when we standardized by area (Figure 8(4)) show almost no difference between the two data sets. That is, when conducting a comparative analysis between two districts that differ substantially in road density and road data precision, standardization by area is preferable.

In addition, taking the district centered on Shinbashi (Figure 9) as the subject of analysis, we determined the cross-correlation function between office buildings and residential facilities based on time intervals and by assuming the presence of automobile users (Figure 10). The automobile speed of travel was defined as shown in Figure 10. From the results of the analysis based on spatial distance, we can see that the locations of the office and house have a spatial lag of 2 km. From the perspective of the automobile user, we can see that the two facilities have two temporal lags of approximately 4 and 7 minutes.

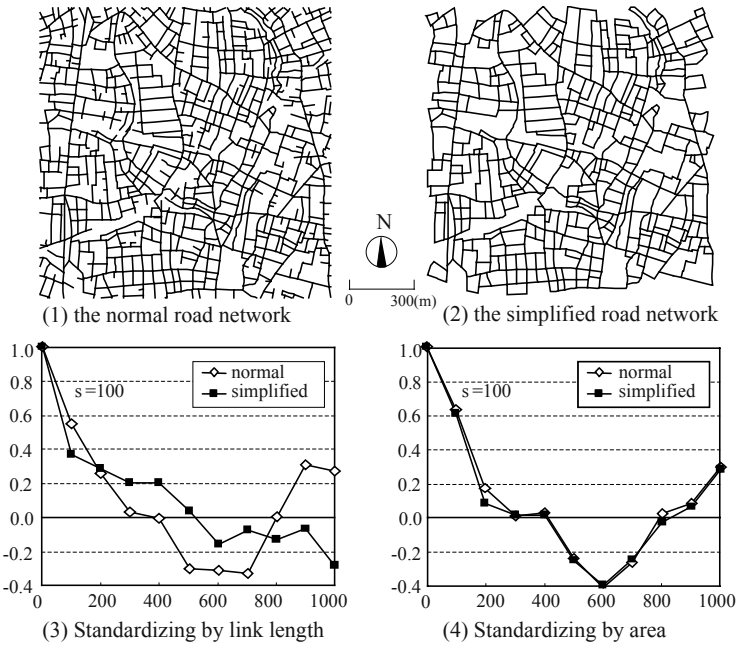
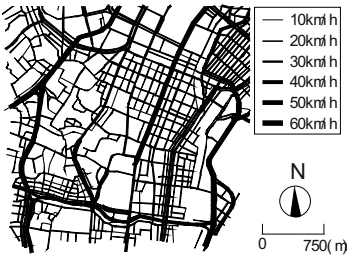
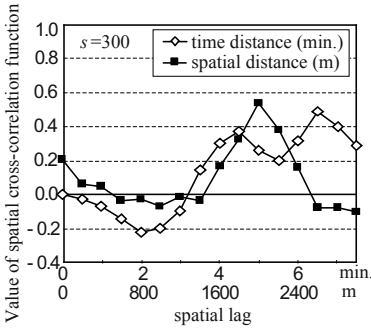


Fig. 8. Influence due to road density on spatial correlation function

Fig. 9. Study area (Shinbashi in Tokyo)



The automobile speed of travel is defined by the legally permitted speed. In case it's not specified, defined by the width of roads.
30 km/h (13 m or more),
20 km/h (5.5 - 13 m),
10 km/h (less than 5m)



Spatial cross-correlation function of office buildings and residential facilities

Fig. 10. Influence due to the definition of distance

6 Summary and Conclusions

In this study, we examined methods for extending conventional spatial correlation analysis, defined in terms of a Euclidean space, to a network space. First, we discussed the differences between spatial correlation defined in a two-dimensional continuous space and a network space, and mathematically defined spatial correlation function on network space as a function of space lag. Next, we developed a method to get re-sampling data by random-point-approach, since it is very difficult to divide network spaces into cells such that each length is made equal, in the same way that a two-dimensional space is divided into a cell. Next, we examined variations of different calculations according to characteristics of data to be analyzed and purpose of analysis in which we use time-distance instead of network-distance, the continuous variables instead of discrete variables and so on. Furthermore, we attempted a verification analysis using data relating to the spatial distribution of real urban facilities and assessed the effectiveness of the proposed method. Namely, we demonstrated the effects of network distance on the resulting spatial correlation when compared to Euclidian distance. Also we pointed out that the variables should be standardized by the area of regions created by dividing the line Voronoi diagram by the edges of line-buffers, since the value of spatial variables standardized by the length of line-buffers will differ according to the road density.

Acknowledgements

The authors would like to acknowledge the valuable comments and useful suggestions from anonymous reviewers to improve the content and clarity of the paper.

References

Aoki Y (1986a) Space Correlation Analysis as A Method of Mesh Data Analysis: Part 1 Problems on mesh data analysis and theory of space correlation analysis (in Japanese). Journal of Architectural Planning and Engineering 364: 94-101

Aoki Y (1986b) Space Correlation Analysis as A Method of Mesh Data Analysis: Part 2 Application on measuring of spatial-continuity, coexistence, exclusion of land use (in Japanese). Journal of Architectural Planning and Engineering 364: 119-125

Aoki Y (1987) Space Correlation Analysis as A Method of Mesh Data Analysis: Part3 On the applicability of the spatial influence function model (in Japanese). Journal of Architectural Planning and Engineering 364: 29-35

Berry B J, Marble D F (1968) Spatial Analysis: A Reader in Statistical Geography, Prentice-Hall

Black W R (1992) Network Autocorrelation in Transport Network and Flow Systems, Geographical Analysis 24: 207-222

Black W R, Thomas I (1998) Accidents on Belgium's motorways: a network autocorrelation analysis, Journal of Transport Geography 6(1): 23-31

Clark P J, Evans F C (1954) Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations, Ecology 35: 445-453

Clark P J, Evans F C (1955) On Some Aspects of Spatial Patterns in Biological Populations, Science 121: 397-398

Cliff A D, Ord J K (1973) Spatial Autocorrelation, Pion

Fotheringham A S (2009) "The Problem of Spatial Autocorrelation" and Local Spatial Statistics, Geographical Analysis 41: 398–403

Getis A (2008) A history of the concept of spatial autocorrelation: a geographer's perspective, Geographical Analysis 40: 297-309

Goodchild M F (2009) What Problem? Spatial Autocorrelation and Geographic Information Science, Geographical Analysis 41: 411–417

Griffith D A (2003) Spatial Autocorrelation and Spatial Filtering: Gaining Understanding Through Theory and Scientific Visualization, Springer-Verlag, Berlin, Germany

Griffith D A (2009) Celebrating 40 Years of Scientific Impacts by Cliff and Ord, Geographical Analysis 41: 343–345

Kitamura M, Okabe A (1995) A method for Estimating Market Areas on a Network (in Japanese). Theory and Applications of GIS 3 (1): 17-24

Koshizuka T, Kobayashi J (1983) On the Relation between Road Distance and Euclidean Distance (in Japanese). Papers on City Planning 18: 43-48

Leenders R A J (2002) Modeling social influence through network autocorrelation: constructing the weight matrix, Social Networks 24: 21-47

Okabe A, Boots B, Sugihara K, Chiu S N (1992) Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, John Wiley & Sons, Chichester, United Kingdom.

Okunuki K, Shiode S, Okabe A, Okano K, Kaneko T (2005) SANET version 3 (in Japanese). Papers and Proceedings of the Geographic Information Systems Association 14: 337-340

Okabe A, Satoh T, Sugihara K (2009) A kernel density estimation method for networks, its computational method and a GIS-based tool (in Japanese). International Journal of Geographical Information Science 23 (1): 7-32

Peeters D, Thomas I (2009) Network Autocorrelation, Geographical Analysis 41: 436-443

Ripley B D (1981) Spatial Statistics, John Wiley & Sons, New York

Roach S A (1968) The theory of Random Clumping, Methuen, London

Shiode S, Okabe A (2004) Analysis of Point Patterns Using the Network Cell Count Method (in Japanese). Theory and Applications of GIS 12 (2): 155-164

Yomono H (1993) The Computability of the Distribution of the Nearest Neighbour Distance on a Network (in Japanese). Theory and Applications of GIS 1: 47-56

Yamada I, Okabe A (2000) The K Function Methods on a Network (in Japanese). Theory and Applications of GIS 8 (1): 75-82

GIS-based Map-matching: Development and Demonstration of a Postprocessing Map-matching Algorithm for Transportation Research

Ron Dalumpines, Darren M. Scott

TransLAB: Transportation Research Lab, School of Geography & Earth Sciences, McMaster University, Hamilton, Ontario, Canada
dalumprf@mcmaster.ca, scottdm@mcmaster.ca

Abstract. This paper presents a GIS-based map-matching algorithm that makes use of geometric, buffer, and network functions in a GIS – to illustrate the suitability of a GIS platform in developing a postprocessing map-matching algorithm for transportation research applications such as route choice analysis. This algorithm was tested using a GPS-assisted time-use survey that involved nearly 2,000 households in Halifax, Nova Scotia, Canada. Actual routes taken by household members who travelled to work by car were extracted using the GPS data and the GIS-based map-matching algorithm. The algorithm produced accurate results in a reasonable amount of time. The algorithm also generated relevant route attributes such as travel time, travel distance, and number of left and right turns that serve as explanatory variables in route choice models. The ease and flexibility of the Python scripting language used in developing the GIS-based map-matching algorithm make this tool easy to develop and implement. It can be improved to suit data inputs and specific fields of application in transportation research. As GIS increasingly becomes a popular tool for transportation, it is timely to exploit GIS as a platform for developing postprocessing map-matching algorithms for transportation research.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_6,

1 Introduction

The increasing popularity of Global Positioning Systems (GPS) inspires some renewed interests in travel behavior research. Person-based GPS devices are increasingly used in travel or time use surveys (Doherty 2001; Murakami and Wagner 1999; Ogle et al. 2002; Wolf et al. 1999; Casas and Arce 1999; Yalamanchili et al. 1999; Draijer et al. 2000; Pearson 2001; Wagner 1997). Matching the GPS coordinates to the digital road network has become an accepted approach in determining the actual routes taken by travelers, thus improving travel behavior analysis by providing a more accurate account of observed routes (Chung and Shalaby 2005; Marchal et al. 2005; Schuessler and Axhausen 2009). This approach is commonly known as map-matching in the field of car navigation and transportation research. Map-matching is a method of tracing the path or route taken by a traveler (represented by a sequence of GPS points) relative to a digital road network map. The underlying issues in map-matching has been extensively explored within the broader field of geographic information science focusing on different applications: geographic integration (Devogele 2002; Harvey 1994, 2005; Harvey and Vauglin 1996a, 1996b; Walter and Fritsch 1999), similarity measures for feature/geographic data matching (Bel Hadj Ali 1997; Lemarié and Raynal 1996; Vauglin and Bel Hadj Ali 1998), spatiotemporal databases, and moving objects data (Brakatsoulas et al. 2005; Cao and Wolfson 2005).

Map-matching can be classified generally into real-time and postprocessing map-matching (Quddus et al. 2007). Real-time map-matching captures the location of a traveler in the road network with a real-time feed of GPS locations (often augmented by data from dead reckoning devices). Postprocessing map-matching takes GPS data recorded from a travel or time-use survey and matches it to the road network to trace the routes taken by travelers. This postprocessing procedure allows the integration of network attributes with the socio-economic information of travelers, providing data that can be used for analysis and model estimation.

Although most map-matching approaches use geometric and topological analysis, two common built-in functions in most GIS packages, very limited studies have attempted to develop a postprocessing map-matching algorithm in a GIS platform because most published articles on map-matching focus on real-time navigation applications (Quddus et al. 2007) and the perceived slow performance of map-matching in a GIS (Schuessler and Axhausen 2009). In the case where it is used for real-time map-matching, GIS use is limited only to visualize the map-matching results (Taylor et al. 2006). Few studies have been published on postprocessing

map-matching algorithms specifically for transportation research. At least 35 articles on map-matching are claimed to be published for the period 1989-2006 (Quddus et al. 2007), although extensive literature exists in geographic information science pertaining to a similar concept but used for different applications (e.g. Brakatsoulas et al. 2005; Cao and Wolfson 2005). Yet in the context of transportation research, to the authors' knowledge, only two articles are written on postprocessing map-matching developed and implemented in a GIS platform (Chung and Shalaby 2005; Zhou 2005). The large majority of the articles focus on real-time map-matching algorithms generally developed for navigation purposes. This suggests the lack of research for postprocessing map-matching algorithms and the need for more of these tools for transportation research, particularly in travel behavior analysis. This paper helps to fill this gap by introducing a postprocessing map-matching algorithm developed in a GIS platform to support travel behavior research in exploiting the increasing popularity of GPS in travel or time-use studies. Hence, this paper argues that a GIS is an ideal platform for the development of a postprocessing map-matching algorithm for transportation research.

The research presented in this paper is unique in two respects: technique and data input. Compared to previous map-matching algorithms, this is the first purely GIS-based map-matching algorithm for postprocessing person-based GPS data. Only two other published studies used GIS as a platform in developing postprocessing map-matching algorithms; however, they employed real-time map-matching procedures such as a reconfiguration of Greenfeld's (Greenfeld 2002) weighted topological algorithm (Chung and Shalaby 2005) and multiple-hypothesis testing matching with rank aggregation (Zhou 2005). The algorithm presented in this paper utilizes mainly built-in functions in a GIS such as buffer analysis and route analysis tools. In terms of data input, the algorithm uses the largest GPS-assisted time use survey undertaken to date (Bricka 2008). This research is a novel attempt to extract observed routes for work trips using GPS data and time diaries (episode data file). This is different from the two related studies on postprocessing GPS data. The first used large GPS records without any additional information (Schuessler and Axhausen 2009). The second had travel survey data but needed to re-enact the trip data using a person-based GPS (Chung and Shalaby 2005). Also, the proposed algorithm fully utilizes the network dataset from a private data provider (DMTI) that includes attribute information such as turn restrictions, one-way street information, road classification, road speed, etc. Such effective use of a network dataset in a GIS platform for postprocessing map-matching has not been done before.

The GIS-based map-matching algorithm generates the actual routes taken by respondents based on the GPS data and time diary. The actual routes

(or observed routes) serve as the dependent variable in route choice modeling. Aside from the observed routes, the algorithm also generates a travel time, route distance, and number of left and right turns for each observed route — used as independent variables in route choice models.

2 Introducing GIS platform for postprocessing map-matching

A map-matching problem is characterized by two objectives: 1) identify the link traversed by the traveler and 2) find his/her actual location within that link (Quddus et al. 2007; White et al. 2000). Postprocessing map-matching algorithms focus only on the first objective while real-time map-matching algorithms need to address the two objectives. Postprocessing and real-time map-matching algorithms also differ in data inputs. Road network map and GPS data are often enough for postprocessing map-matching. Real-time map-matching requires other data (e.g. from dead reckoning devices, elevation models, etc.) usually to augment the inaccuracies of GPS in urban environments. The kind and nature of these data inputs and the purpose of the algorithm largely influence the development of the map-matching procedures. For example, postprocessing procedures can create a polyline feature from the entire series of GPS points, which are already available, and match this line to the road network (i.e. global map-matching procedure (Lou et al. 2009)). This is not possible in real-time map-matching because the map-matching needs to process the GPS coordinates as they are being updated online (i.e. incremental map-matching procedure).

2.1 Postprocessing map-matching merits a different approach

Adopting procedures originally developed for real-time map-matching to postprocessing map-matching restricts the search for more appropriate procedures specifically for postprocessing map-matching. For example, the shortest path algorithm has not been used for real-time map-matching but can be appropriately used for postprocessing map-matching applications. Zhou (2005) cites that the shortest path algorithm can be utilized for post-processing map-matching algorithm but did not proceed on exploring the idea. Hence, the core element of the GIS-based postprocessing map-matching as proposed in this paper, which is the use of the shortest path algorithm, is not a new idea. However, this idea, to the authors' knowl-

edge, has not been fully explored particularly in a time when advances in GIS platforms offer more flexibility and advanced functionality.

Furthermore, there should be a clear distinction as far as postprocessing map-matching is concerned. For this reason, the development of postprocessing map-matching algorithms should take a different approach from those of real-time map-matching. The dominance of the real-time map-matching procedures in the literature leads to the on-going adoption of these procedures to postprocessing applications. Since real-time map-matching procedures have not established an affinity with GIS platforms, postprocessing map-matching procedures currently focus on developing procedures in non-GIS platforms.

2.2 Premise to the use of the GIS platform for map-matching

The platforms used for the development of the map-matching algorithms reveal the range of techniques that can be employed. GIS provides excellent data models and tools in dealing with spatial data. For example, ArcGIS® provides an advanced network data model that allows for the modeling of complex road layouts by taking into account road design parameters such as turn restrictions, road hierarchy, and impedances. This strength of GIS makes it an ideal platform in developing a postprocessing map-matching algorithm that fully integrates network topology and attributes to match streams of GPS points to the road network.

However, GIS packages are often proprietary, comprehensive, and platform-dependent. For these reasons, most map-matching procedures are developed and implemented in non-GIS platforms. For example, Java is free, platform-independent, and used in some postprocessing procedures (Marchal et al. 2005; Schuessler and Axhausen 2009). Even so, non-GIS platforms have limited capability in handling spatial data models such as road networks and thus rely on a planar network that consists of nodes and arcs (links). This paper provides evidence that a postprocessing map-matching algorithm that utilizes a GIS network data model is effective in integrating topological information and resolving the map-matching problems of complex road intersections.

3 Constraints and limitations of existing algorithms

The existing literature identifies the constraints and limitations of the current map-matching algorithms for transportation applications (Quddus et al. 2007; White et al. 2000). Although the literature review by Quddus et

al. (2007) focuses on real-time map-matching algorithms, some of the major constraints and limitations they identified also apply to postprocessing map-matching algorithms. These constraints and limitations refer to problems associated with the identification of initial links, calibration of threshold values used in decision processes, and the difficulty in correctly matching locations in complex road layouts (e.g. cloverleaf interchanges, flyovers).

3.1 Problem with the initial map-matching process

One of the problems of existing map-matching algorithms is the identification of the initial link. The existing map-matching techniques use an error ellipse or circle to snap the GPS point/s to the junction (intersection) node to identify the initial link. The problem occurs if the junction node falls outside this error region. Moreover, the entire length of the link is assumed to be traversed once an initial link is identified. This is problematic for trip ends covering only a portion of the road link because the travel distance will be overestimated if computed based on all the links traversed. This problem is avoided when using the GIS-based map-matching algorithm. This algorithm snaps the initial GPS point to the nearest link instead of the junction node and the route length is calculated from the portion of the link covered by the GPS trajectory.

3.2 Calibration of threshold values

All of the existing map-matching algorithms use some parameters. For example, a postprocessing map-matching algorithm developed by Marchal et al. (2005) depends on two parameters, N (number of candidate paths) and α (u-turn parameter). The number of parameters increases as the map-matching algorithm becomes complicated. Often it is difficult to recommend default values and this becomes an issue when applying the map-matching technique to a different operational environment (Quddus et al. 2007). A map-matching algorithm that uses a minimum number of parameters that can be calibrated easily will be helpful to transportation researchers.

3.3 Problems at complex intersections

Development of map-matching algorithms should effectively address the problems that arise at intersections. Route changes occur at intersections

making it difficult for map-matching processes to identify the next link (White et al. 2000). This difficulty is more pronounced in complex intersections (e.g. cloverleaf interchanges, flyovers) and further exacerbated by GPS errors. For this reason, the existing literature repeatedly suggests the integration of network topology in map-matching procedures (Marchal et al. 2005; Quddus et al. 2007; White et al. 2000; Quddus et al. 2003). Most of the existing algorithms did not go beyond the simple connectivity rules, often incorporated in some scoring procedures (or set of rules) to determine the next link after the intersection. Hence, Quddus et al. (2007) recommends the use of road design parameters (e.g. turn restrictions, road classification, etc.) and Marchal et al. (2005) hinted at the use of turn rules to improve map-matching performance particularly at intersections.

To the authors' knowledge, no map-matching algorithm has taken full advantage of these road attributes. Quddus et al. (2007) attributed this to unavailability of data but it can be argued that the standard network data model (planar network model) limits the inclusion of road attributes. Spatial road network data are available from governments for free and private providers sell more comprehensive data at a reasonable cost. Private data vendors provide route logistics or road network data in GIS formats (e.g. ArcGIS®, MapInfo®), which the existing map-matching methods have not taken full advantage.

Therefore, this paper argues that a GIS platform should be used in developing a postprocessing map-matching algorithm to utilize the network topology and road attributes that can be handled easily in a GIS environment. The next section describes the GIS-based postprocessing map-matching algorithm followed by the testing results using the Halifax STAR Project dataset (focusing on routes taken by 104 individuals during their drive to work in the morning).

4 GIS-based map-matching algorithm

The core of the GIS-based map-matching algorithm is the use of a route analysis tool in ArcGIS® (Network Analyst extension). This tool uses a shortest path algorithm and basic inputs (stops, barriers) to generate the shortest path or route. Stops and barriers are the basic parameters that need to be set by the user. Stops refer to the origin and destination locations (i.e. trip origin and destination) used by the shortest path algorithm to generate the best or shortest route. Barriers play a significant role in controlling the shortest path algorithm to generate only the route based on the streams of GPS points that represent a trip. The algorithm creates a buffer region

around GPS trajectories to produce a set of barriers that control the route analysis tool to correctly generate the observed routes. These routes are automatically stored in a file geodatabase feature class format that contains relevant attributes for the route choice analysis (e.g. travel distance, travel time, number of left and right turns). These attributes are automatically added by the algorithm to every route generated. The route generated depends on the impedance or cost defined by the user. Travel time is the default impedance used by the algorithm but the user can change it to travel distance if desired.

The postprocessing GIS-based algorithm is developed and implemented in ArcGIS® v9.3.1 using Python scripting language. Python scripting is free and well supported in ArcGIS® and works well with ArcObjects™ - the building blocks of ArcGIS® software. The algorithm can be run as a standalone program or added as a tool in ArcGIS®. The standalone implementation saves some processing overhead and hence it has computational speed advantage. The latter approach provides a user-friendly GUI, allowing users to specify the input data and parameters. To run the algorithm via GUI (Figure 1), the user specifies the following parameters: the workspace location of the file geodatabase containing the GPS data, a sample GPS data file from the file geodatabase (for the script to read the attribute fields of the GPS data file), line field and sort fields used to convert the GPS points into a polyline feature, the network dataset, and the buffer distance in meters (50 m is the default value).

Python scripting is used to automate the detailed steps for the GIS-based map-matching, the steps are described as follows:

1. Convert the stream of GPS points (representing the trip made by a traveler) into a polyline feature. The first and the last GPS points in the sequence are designated as stops (origin and destination points) in the network analyst module in ArcGIS®. The intermediate points between the stops are used to generate the polyline feature; this feature is the basis for the buffer in step 2.
2. Create a buffer around the polyline feature based on user-defined distance. The buffer distance should be, more or less, 5x to 6x the horizontal accuracy of the GPS data. This is based on the results of the sensitivity analysis, which is explained further in the next section on results (section 5.4). The experiment for GPS data with a horizontal accuracy of 10 m revealed that a buffer of 50 m produces accurate results. This was set as the default distance in the algorithm but can be changed by the user.

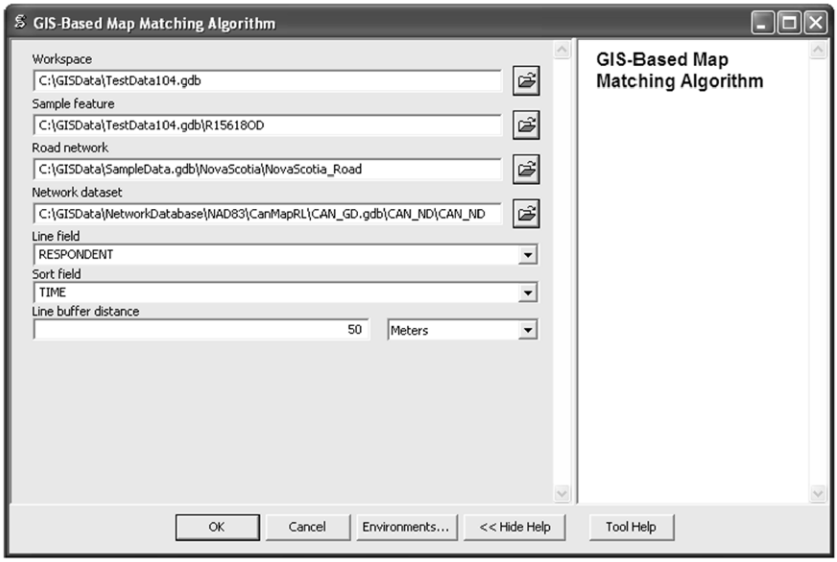
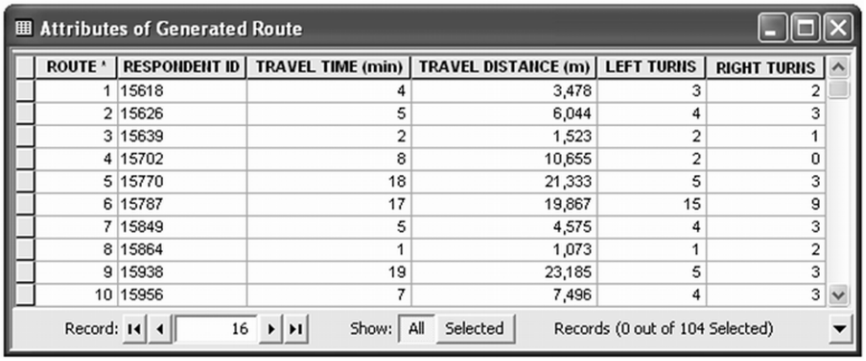


Fig. 1. Sample GUI of the GIS-based map-matching algorithm showing the input data and the buffer distance parameter

3. Assign the stops and barriers for the route solver (from ArcGIS® Network Analyst module). Start and end points define the stops. (The route solver can also work with multiple stops in between the start and end points, similar to the Traveling Salesman Problem (TSP)). Barriers are defined by the intersection of the boundary of the buffer region created in the previous step and surrounding links. Barriers ensure the accuracy and efficiency of the shortest path algorithm. This step assumes that there are no errors particularly gaps in the GPS data. Outliers and other errors are handled by the GPS data preprocessing module.
4. The observed route is generated (or not generated) depending on the buffer distance specified by the user. This route is the shortest or best route generated by the shortest path algorithm (using the origin and destination points) inside the buffer region. Topological rules and road attributes (e.g. one-way restrictions, road hierarchy, etc.) are used by the built-in shortest path algorithm in ArcGIS® in generating the shortest path between origin and destination points.
5. The network attribute table is updated for the number of left and right turns in traversing the observed route, aside from the travel distance and travel time that are automatically generated by the route solver (Figure 2).



Attributes of Generated Route

ROUTE *	RESPONDENT ID	TRAVEL TIME (min)	TRAVEL DISTANCE (m)	LEFT TURNS	RIGHT TURNS
1	15618	4	3,478	3	2
2	15626	5	6,044	4	3
3	15639	2	1,523	2	1
4	15702	8	10,655	2	0
5	15770	18	21,333	5	3
6	15787	17	19,867	15	9
7	15849	5	4,575	4	3
8	15864	1	1,073	1	2
9	15938	19	23,185	5	3
10	15956	7	7,496	4	3

Record: 16 Show: All Selected Records (0 out of 104 Selected)

Fig. 2. Portion of the attribute table generated by the GIS-based map-matching algorithm showing the important attributes for each observed route (i.e. travel time (minutes), travel distance (meters), and the number of left and right turns) relevant to route choice modeling.

5 Results

5.1 Data input and preprocessing

The Halifax Space-Time Activity Research (STAR) Project was claimed to be the world's first largest GPS-assisted prompted-recall time diary survey (Bricka 2008). The survey was conducted for a 2-day period covering approximately 2,000 households in Halifax, Nova Scotia, Canada from 2007 to 2008. Person-based GPS devices were used. The GPS data have a spatial resolution of within 10 meters (but generally <3m) and a temporal resolution of 3 recordings every 2 seconds. About 47 million GPS points were collected. The GPS data were obtained in SPSS format from the Halifax STAR Project then converted into GIS format as point features. Start time and end time corresponding to trip ends for work trips were extracted from a time diary episode data file into a matrix of respondent IDs, start time, and end time. The matrix was used to extract the portion of the daily trips corresponding to work trips by car using a Python script in ArcGIS®. Work trips by car were extracted because most individual daily trips consist of this kind of trip. Moreover, work trips are extensively studied in the field of transportation, particularly in route choice modeling. The selection of the sample is motivated by the potential application of the GIS-based algorithm to generate the input data for route choice modeling. Thus, the selec-

tion focused on interzonal, home-based work trips that are at least a kilometer in length, and performed by unique individuals. Out of 3,023 simple work trips from the STAR time diary - episode data file, about 574 home-based work trips are selected. Some of the reported work trips in the episode data file have missing GPS trajectories. All the GPS trajectories representing home-based work trips are preprocessed. Data preprocessing involved removal of outliers and gaps. GPS points with horizontal dilution of precision (HDOP) value greater than 2 are removed. Also, "position jumps" are removed if the calculated speed between two consecutive GPS points exceeds 50 m/s (Schuessler and Axhausen 2009). Gaps are filled in using proximity analysis and data management tools in ArcGIS®. After data preprocessing, 104 work trips are finally selected.

A first experiment is performed on the sample of 104 work trips that accounts for about one percent of the data (46K points) or about 18 percent of total home-based work trips. Each work trip from the sample begins at home and ends at the work place. The small sample was chosen because it is easy to manage and helps facilitate the manual validation of the routes generated by the algorithm – enough to illustrate the performance of the GIS-based map-matching algorithm. Future experiments will attempt to use the entire GPS dataset, starting with the application of the algorithm to extract routes for the 3,023 simple work trips. The validation process involved visual checking of home and work place locations and GPS trajectories with the aid of contextual information from time diary data and satellite imagary. The preprocessed GPS data were stored in an ArcGIS® file geodatabase ready for map-matching, representing about 104 individual work trips.

The GIS-based map-matching algorithm requires two inputs: (1) the preprocessed GPS data file stored in file geodatabase format and (2) the network dataset. The preprocessed GPS data for 104 individual work trips comprise about 440 points per trip. The network dataset was from DMTI Spatial CanMap® Route Logistics Version 2008.3 that provides a detailed road and highway network for Canada. A subset of this network was extracted for Nova Scotia because some of the trips go beyond the Halifax region. The road network for Nova Scotia consists of 116,647 links and 98,132 junctions.

5.2 Accuracy

The algorithm correctly generated the routes for 88 percent of the work trips (91 routes). The few inaccuracies are mainly attributed to the wrong turn restrictions in the network dataset from DMTI. Manual correction of

the wrong turn restrictions produced accurate results. However, the algorithm performed well at complex intersections (Figure 3). The validation is performed by visually retracing the routes taken by respondents using time diary records for each of the 104 respondents. The advanced network data model and the shortest path algorithm in the GIS platform effectively utilized the network topological information enabling the GIS-based algorithm to produce accurate results. Other map-matching algorithms that use the planar network model have difficulty in matching GPS trajectories at complex intersections (Quddus et al. 2007). This is because of the limited capability of the planar network in modeling complex road layouts.

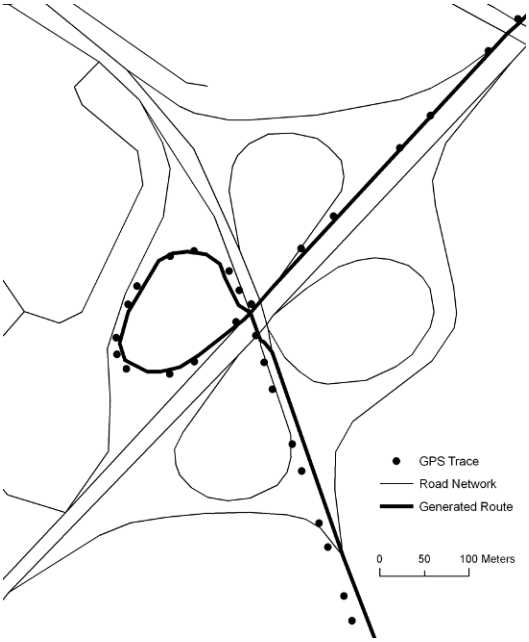


Fig. 3. An example of a complex intersection where the GIS-based map-matching algorithm accurately generated the route for the GPS trace

Unlike the planar network data model commonly used in existing map-matching algorithms (Marchal et al. 2005; White et al. 2000; Quddus et al. 2003), the proposed map-matching uses an advanced network data model of ArcGIS®. The planar network data model is a simple representation of road network in terms of nodes and arcs. It is computationally efficient but very limited in making use of topological rules and road attributes to model the actual road network. For this reason, White et al. (2000), Quddus et al. (2003), and Marchal et al. (2005) call for the integration of network topology to improve accuracy and address the limited capability of map-

matching when it comes to complex road layouts, particularly at road intersections. The network dataset in ArcGIS® is an advanced network data model that fully utilizes connectivity rules and road attributes (e.g. costs, restrictions, road classification) that allow for the modeling of complex scenarios. The route analysis tool makes use of the ArcGIS® advanced network data model and has a potential in addressing the two prevailing issues in the literature: effective use of topological information and dealing with the problems that arise in road intersections. The route analysis uses the shortest path algorithm to solve for the best route based on the cost or impedance parameter. This produces accurate results in a matter of seconds. This is made possible with the use of the advanced network connectivity and attribute data model.

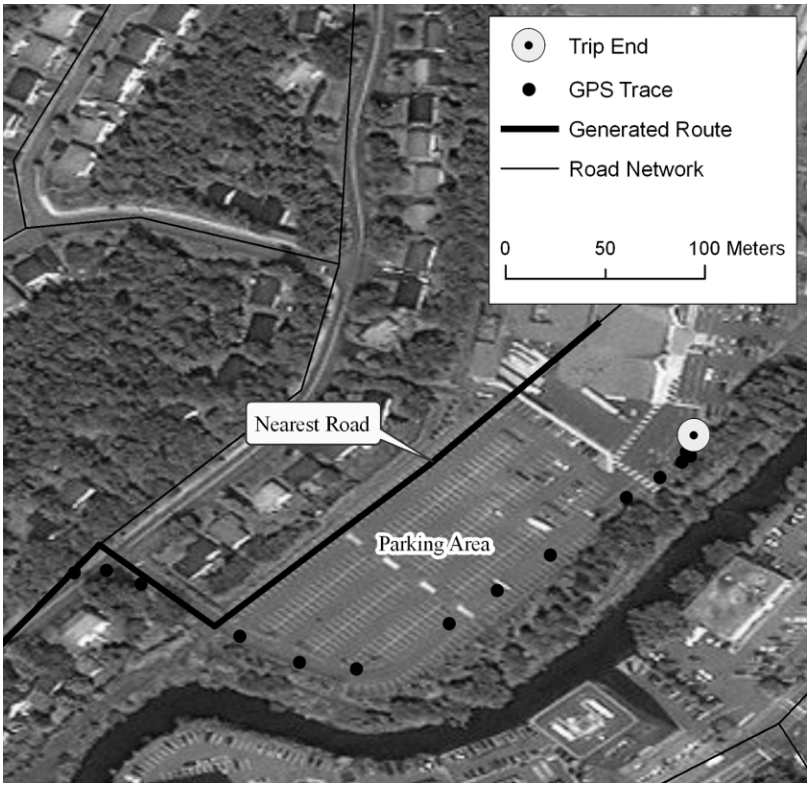


Fig. 4. The algorithm sticks to the nearest road whenever some network link is missing

The GIS-based map-matching algorithm avoids the problem with the initial map-matching process. The snapping function in the GIS environment

works effectively in snapping the initial GPS point to the nearest road link. However, the resolution or the quality of the road network often affects the accuracy of the algorithm at the start and end locations. Access roads that connect parking areas or home locations to main roads are missing in most digital road networks. At the start or end of the trip, the GIS-based map-matching uses the nearest road to match the GPS trajectories when access roads are missing (Figure 4). Based on the experiment results, the missing access roads to home locations or parking areas account for about a 10 percent difference between the actual trip distances and generated routes. An in-depth analysis of problems associated with missing links has not been fully addressed here but would be interesting to investigate in the future.

5.3 Computing speed

The testing of the algorithm is implemented in a PC with an Intel Core Duo processor clocked at 2.66 GHz with 3 gigabytes of physical memory. The experiment revealed an average computing speed of one minute per trip. This is approximately 6 seconds per point relative to the sample. The computing speed per trip seems to remain constant regardless of an increase in the number of trips or route length (Figure 5). However, computing speed gradually increases with the increase in buffer distance or the increase in complexity of the GPS trajectory that prolongs the creation of the buffer region. Chung and Shalaby (2005) reported a computing speed of 2-6 minutes per trip in a PC with an Intel Pentium III 1 GHz. No direct comparison can be made with other previous studies because of the lack of objective and comparable performance indicators. The computing speed can be improved by minimizing the overheard processing through efficient coding.Although considered important, computing performance is not the top priority for postprocessing map-matching for transportation research. However, the GIS-based map-matching algorithm demonstrated an acceptable computational speed in generating accurate routes for 88 percent of the work trips tested.

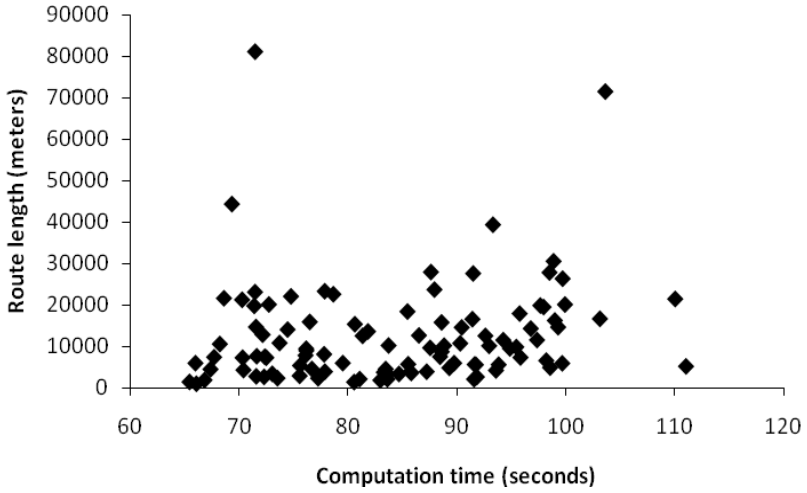


Fig. 5. Plot of the computation time over route length for a sample of 104 routes

5.4 Advantages and limitations

The accuracy of the GIS-based map-matching algorithm is sensitive to the buffer distance. A sensitivity analysis is conducted on some randomly selected trips. These trips are selected because they are more complex than the rest, often with loops and sharp curves. Buffer distances of 10 m, 15 m, 20 m,…, 100 m are tested. The results show that no routes are generated for buffer distances below 50 m. Inaccurate routes are generated for buffer distances of 60 m and above. Figure 6 shows the effect of buffer distances to the map-matching accuracy. Therefore, the buffer distance for the GPS trajectory should be, more or less, 5x to 6x the horizontal accuracy of GPS data. This range of buffer distance values accounts for the width of the roads, the sharpness of curves, and GPS positioning errors. Values greater than this threshold will cover irrelevant links resulting in generating incorrect routes; values lower than the threshold will be too restrictive and no shortest path or route will be generated. The buffer distance that will produce accurate map-matching results depends on the complexity of the road network and the horizontal accuracy of the GPS device. This distance can be easily set by the user unlike some threshold values in other map-matching algorithms that need in-depth empirical study to determine the appropriate values for several parameters (Marchal et al. 2005; Quddus et al. 2003). Future research should perform a more thorough investigation

concerning the buffer distance parameter and computing performance (e.g. using different GPS and road network datasets).

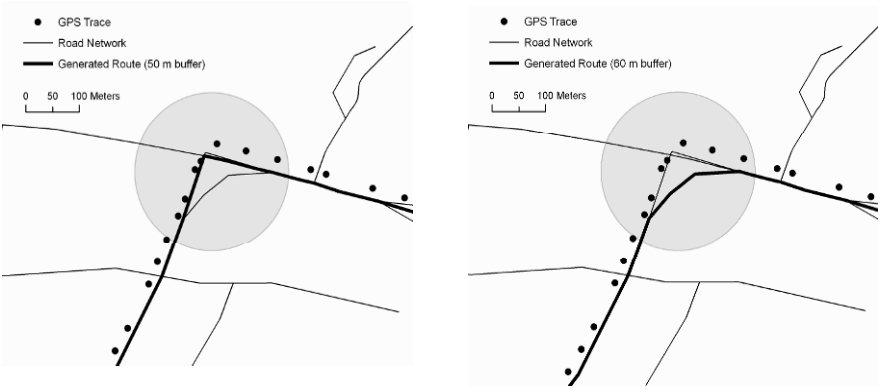


Fig. 6. Sensitivity of map-matching algorithm to buffer distance: (a) buffer distance = 50 m and (b) buffer distance = 60 m

The shortest path algorithm in ArcGIS® is the core component of the map-matching algorithm proposed in this research. The shortest path algorithm alone is computationally efficient in generating routes and could also take into account multiple stops or destinations. With some modification, the algorithm can be extended or expanded to automatically extract multi-modal trips, similar to the trip reconstruction tool (Chung and Shalaby 2005), or a GPS postprocessing tool (Schuessler and Axhausen 2009).

The GIS-based algorithm is timely for the increasing availability of road network data from government sources and private data vendors. Rich network datasets of good quality are readily available, mostly from private data vendors for a reasonable price. This reduces the time to provide road network data required for the map-matching algorithm. In the absence of road network data, new data can be created in the GIS environment.

In summary, the advantages of the proposed algorithm are the simple user interface (GUI), parameters can be changed to suit the demands of a particular dataset (i.e. using the appropriate buffer distance), can be expanded to perform more functions (e.g. extract multi-modal trips), generates accurate routes within a reasonable amount of time, and its portability. The algorithm was developed using Python script and can be easily added in ArcGIS® as a tool. The portability of the script makes it available to many users. Also, the script can be easily edited and improved by accessing the script file in any text reader application.

6 Conclusion

This paper argues that a GIS is the ideal platform for the development of postprocessing map-matching algorithm for transportation research like route choice modeling because it is easier to develop and implement, is scalable, and generates accurate results at an acceptable computing cost. To support this argument, this paper presented a postprocessing algorithm developed and implemented in a GIS platform. The development of the algorithm is easy and fast by making use of the functionalities already available in commonly used GIS platforms. As shown in this paper, the GIS-based map-matching algorithm is able to deal effectively with complex road intersections and generate accurate routes at reasonable computing costs. The script can be improved and can be easily employed by researchers with GIS in their research environment. Basically, the algorithm makes use of buffer and network analysis that can effectively be done in GIS. The increasing availability of commercially available network datasets and GPS-assisted time use or activity surveys provide a timely basis for this kind of algorithm. This algorithm can be easily tailored to the needs of researchers in analyzing route choice behavior.

However, several issues need to be resolved for further improvement of the algorithm. Computing speed can be improved by the use of efficient coding and moving computing intensive processes to a faster programming language like C++. Seamless integration with the GPS data preprocessing is needed and this is another research direction that the authors will undertake. This integration may also include the development of a new module that will enable users to easily link GPS data with a time diary episode data file or travel survey data to enable extraction of reported trip ends, travel time, and other information.

The GIS-based map-matching algorithm can be expanded to automatically detect and extract trips made by other modes such as public transportation, walking and cycling. The development of this trip reconstruction tool is perfectly suited for the Halifax STAR dataset and the authors are currently working towards this direction by utilizing GIS as a development platform. Moreover, the GIS-based map-matching algorithm presented in this paper is part of an on-going effort to develop a GIS-based toolkit for route choice modeling.

Acknowledgements

Financial support for this project was provided by a grant awarded to Darren M. Scott from the Natural Sciences and Engineering Research Council of Canada (261850-2009). The authors greatly acknowledge suggestions from the three anonymous reviewers in improving the quality of this paper. The authors also acknowledge the Halifax STAR Project for the GPS and time diary datasets used for the development and testing of the map-matching algorithm, David Wynne for the point-to-line script he made available for free, and the GEOmatics for Informed DEcisions (GEOIDE) for the travel grant awarded to the first author to present this work at the 14th AGILE Conference on Geographic Information Science.

References

Bel Hadj Ali, A. (1997) Appariement geometrique des objets géographiques et étude des indicateurs de qualité. Saint-Mandé (Paris), Laboratoire COGIT.

Brakatsoulas, S., Pfoser, D., Salas, R. and Wenk, C. (2005) On Map-Matching Vehicle Tracking Data. VLDB 2005, pp. 853-864.

Bricka, S. (2008) Non-Response Challenges in GPS-Based Surveys, paper at the 8th International Conference on Travel Survey Methods, May 2006, Annecy, France.

Cao, H. and Wolfson, O. (2005) Nonmaterialized Motion Information in Transport Networks. ICDT 2005, pp. 173-188.

Casas, J., and Arce, C. H. (1999) Trip Reporting in Household Travel Diaries: A Comparison to GPS-Collected Data. Presented at *78th Annual Meeting of the Transportation Research Board*, Washington, D.C.

Chung, E., and Shalaby, A. (2005) A trip reconstruction tool for GPS-based personal travel surveys. Transportation Planning and Technology, Vol. 28, No. 5, pp. 381-401.

Devogele, T. (2002) A new merging process for data integration based on the discrete Frechet distance. In *Advances in Spatial Data Handling*, D. Richardson and P. van Oosterom. New York, Springer Verlag, pp. 167-181.

Doherty, S. (2001) Meeting the Data Needs of Activity Scheduling Process Modeling and Analysis. Presented at *80th Annual Meeting of the Transportation Research Board*, Washington, D.C.

Draijer, G., Kalfs, N. and Perdok, J. (2000) Global Positioning System as Data Collection Method for Travel Research. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1719,* TRB, National Research Council, Washington, D.C., pp. 147–153.

Greenfeld, J. S. (2002) Matching GPS observations to locations on a digital map. Papers presented at the 81th Annual Meeting of the Transportation Research Board. CD-ROM. January 2002, Washington, DC.

Harvey, F. (1994) Defining unmoveable nodes/segments as part of vector overlay: The alignment overlay. In *Advances in GIS Research,* T. C. Waugh and R. C. Healey. London, Taylor and Francis, 1, pp. 159-176.

Harvey, F. (2005) Aligning or Matching: Cartographic Perspectives on Geographic Integration. AutoCarto 2005, Las Vegas, NV, ACSM.

Harvey, F. and Vauglin, F. (1996a) Geometric match processing: Applying Multiple Tolerances. The Seventh International Symposium on Spatial Data Handling (SDH'96), Delft, Holland, International Geographical Union (IGU).

Harvey, F. and Vauglin, F. (1996b) Geometric match processing: Applying Multiple Tolerances. In *Advances in GIS Research,* Proceedings of the Seventh International Symposium on Spatial Data Handling, M. J. Krakk and M. Molenaar. London, Taylor & Francis, 1, pp. 155-171.

Lemarié, C. and Raynal, L. (1996) Geographic data matching: First investigations for a generic tool. GIS/LIS '96, Denver, Co, ASPRS/AAG/URISA/AM-FM.

Lou, Y., Xie, X., Zhang, C., Wang, W., Zheng, Y. and Huang, Y. (2009) Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS)*, pp. 544-545.

Marchal, F., Hackney, J. and Axhausen, K.W. (2005) Efficient map matching of large global positioning system data sets. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1935, Transportation Research Board of the National Academies, Washington, D.C., pp. 93-100.

Murakami, E., and Wagner, D.P. (1999) Can Using Global Positioning System (GPS) Improve Trip Reporting? *Transportation Research Part C,* Vol. 7, pp. 149–165.

Ogle, J., Guensler, R., Bachman, W., Koutsak, M. and Wolf, J. (2002) Accuracy of Global Positioning System for Determining Driver Performance Parameters. In *Transportation Research Record: Journal of the Transportation Research Board: No. 1818,* Transportation Research Board of the National Academies, Washington, D.C., pp. 12–24.

Pearson, D. (2001) Global Positioning System (GPS) and Travel Surveys: Results from the 1997 Austin Household Survey. Presented at 8th Conference on the Application of Transportation Planning Methods, April 2001, Corpus Christi, Texas.

Quddus, M. A., Ochieng, W.Y. and Noland, R.B. (2007) Current map-matching algorithms for transport application: State-of-the-art and future research directions. *Transportation Research Part C*, Vol. 15, pp. 312-328.

Quddus, M. A., Ochieng, W.Y., Zhao, L. and Noland, R.B. (2003) A general map matching algorithm for transport telematics applications. *GPS Solutions*, Vol. 7, pp. 157-167.

Schuessler, N. and Axhausen, K.W. (2009) Processing raw data from Global Positioning Systems without Additional Information. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2105, Transportation Research Board of the National Academies, Washington, D.C., pp. 28-36.

Taylor, G., Brunsdon, C., Li, J., Olden, A., Steup, D. and Winter, M. (2006) GPS accuracy estimation using map matching techniques: Applied to vehicle positioning and odometer calibration. *Computers, Environment and Urban Systems*, 30, pp. 757-772.

Vauglin, F. and Bel Hadj Ali, A. (1998) Geometric matching of polygonal surfaces in GISs. ASPRS Annual Meeting, Tampa, FL, ASPRS.

Wagner, D. P. (1997) Lexington Area Travel Data Collection Test: GPS for Personal Travel Surveys. Final Report. Office of Highway Policy Information and Office of Technology Applications, Battelle Transport Division, FHWA, Sept. 1997, Columbus, Ohio.

Walter, V. and Fritsch, D. (1999) Matching spatial data sets: a statistical approach. *International Journal of Geographic Information Science,* 13(5), pp. 445-473.

White, C. E., Berstein, D. and Kornhauser, A.L. (2000) Some map matching algorithms for personal navigation assistants. *Transportation Research Part C*, 8, pp. 91-108.

Wolf, J., Hallmark, S., Oliveira, M., Guensler, R. and Sarasua, W. (1999) Accuracy Issues with Route Choice Data Collection by Using Global Positioning System. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1660,* TRB, National Research Council, Washington, D.C., pp. 66–74.

Yalamanchili, L., Pendyala, R.M., Prabaharan, N. and Chakravarty, P. (1999) Analysis of Global Positioning System-Based Data Collection Methods for Capturing Multistop Trip-Chaining Behavior. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1660,* TRB, National Research Council, Washington, D.C., pp. 58–65.

Zhou, J. (2005) A three-step general map matching method in the GIS environment: Travel/transportation study perspective. UCGIS Summer Assembly 2005. Wyoming. http://www.ucgis.org/summer2005/studentpapers.htm. Last date accessed 07.2010.

Modeling Micro-Movement Variability in Mobility Studies

Dirk Hecker, Christine Körner, Hendrik Stange, Daniel Schulz, Michael May

Fraunhofer IAIS, Sankt Augustin, Germany
{dirk.hecker, christine.koerner, hendrik.stange, daniel.schulz, michael.may}@iais.fraunhofer.de

Abstract. During recent years, the interest in the exploitation of mobility information has increased significantly. Along with these interests, new demands on mobility data sets have been posed. One particular demand is the evaluation of movement data on a high level of spatial detail. The high dimensionality of geographic space, however, makes this requirement hard to fulfill. Even large mobility studies cannot guarantee to comprise all movement variation on a high level of detail. In this paper, we present an approach to increase the variability of movement data on a microscopic scale in order to achieve a better representation of population movement. Our approach consists of two steps. First, we perform a spatial aggregation of trajectory data in order to counteract sparseness and to preserve movement on a macroscopic scale. Second, we disaggregate the data in geographic space based on traffic distribution knowledge using repeated simulation. Our approach is applied in a real-world business application for the German outdoor advertising industry to measure the performance of poster sites.

1 Introduction

One important detail of empirical studies is the assembling of the data sample. Only if the sample captures all relevant variation of the quantity of interest, the associated study will deliver truthful results. Clearly, a data

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_7,

sample must be larger as the variety within the data is increased. In social sciences, the sample size usually does not pose a problem. However, this is not the case for mobility studies. In contrast to socio-demographic variables with typically few categorical values, geographic space and the space of all possible movement paths is vast. Even in discretized form geographic space may range from a few hundred districts to a few thousand communities or a few million street segments for a nationwide observation. In order to analyze mobility on a fine level of detail, very large data samples have to be formed. More concretely for Germany, this would mean to form a representative set of test persons that cover their movements with about 6.7 million street segments in a given period of time. We have evaluated a data set with 42,780 test persons of mixed Global Positioning Technology (GPS) and Computer Assisted Telephone Interviews (CATI) monitoring for up to seven days. Although this is a large mobility data set, the trajectories of the test persons cover barely 26.7% of the German street network. Clearly, this coverage is not sufficient for evaluations at thestreet level. However, as mobility studies are very laborious and expensive, it is not realistic to perform larger studies in the near future with GPS or CATI technology.

So far, researchers and practitioners in the mobility or transportation area have restricted their evaluations to either a) analyzing mobility without spatial references or b) aggregating the data on a broader spatial level. For example, the study *Mobility in Germany 2008* (BMVBS 2010) evaluates variables as the average number of trips or travelled kilometers per day or the chosen means of transportation. However, it does not refer to mobility at the street level. A second example is commuter statistics, which are typically aggregated at the community level as the German *Arbeitswegematrix* by DDS (2010).

In this paper, we present an approach to increase the spatial variability of a given mobility data set while retaining important mobility characteristics of the sample. Thus, it is possible to evaluate mobility information of the sample at the street level. Our approach consists of two parts: a spatial aggregation of the mobility data and a subsequent simulation-driven disaggregation of the data based on information about the traffic distribution. The first step ensures that mobility is preserved on a coarse level of resolution. The second step incorporates background knowledge about the mobile behavior of the population in order to a) stabilize the disaggregation and b) introduce spatial variability to the trajectory data on a fine level of resolution.

Our approach is implemented in a real-world business application for the German outdoor advertising industry. It constitutes an important model part for the evaluation of poster performance. This evaluation is a busi-

ness-critical task in outdoor advertisement, because the pricing of poster sites depends on the performance of the posters. Its significance becomes apparent when considering that the German outdoor advertising market generated net sales of 737.5 million Euros in 2009 (FAW 2010).

Our paper is organized as follows. The next section reviews related work. Section 3 gives a general overview of our approach and all relevant components. Section 4 describes the building of a spatial aggregation system as well as the aggregation method and Section 5 describes the disaggregation method. In Section, 6 we demonstrate our approach for German outdoor advertising and Section 7 concludes the paper.

2 Related Work

Travel surveys are a traditional part of transportation research. However, in recent years many algorithms and visualization techniques for trajectory data have been developed by the data mining and visual analytics community. Also privacy has become an issue in mobility analysis due to the sensitive nature of the data. In this section, we will give a short description of each research direction and delineate our work from existing approaches.

Travel surveys collect information about individual travel behavior. The data is often captured using travel diaries or CATI. Also the extension or substitution of travel surveys with GPS has been approached (Wolf 2003, Wolf et al. 2001). The main focus of travel surveys is, however, the analysis of general movement characteristics on a regional or national level. For example, the study *Mobility in Germany* (BMVBS 2010) which has been commissioned by the German Federal Ministry of Transport, Building and Urban Development, evaluates variables as average travelled kilometers per day, commuting behavior, utilized means of transportation or activities that motivate travelling. The actually taken routes on the street network are not evaluated and are of little importance in travel surveys so far. This is contrary to our analysis approach as we want to rightly represent and evaluate mobility at the street level.

In recent years, trajectory data has drawn the attention of the data mining community. Algorithms have been developed for the clustering of (parts of) trajectories (Rinzivillo et al. 2008, Pelekis et al. 2007, Nanni and Pedreschi 2006), detection of relative motion patterns (Gudmundsson et al. 2007, Laube and Imfeld 2002), or sequential analysis of movement (Zheng et al. 2009, Giannotti et al. 2007, Yang and Hu 2006). However, all these approaches are restricted to the provided trajectory sample. Population movements outside of the traversed geographic space are not part of the

analyses. In our approach, we are interested in increasing micro-movement variability of a given data sample, so that it covers the complete movement space and becomes a better representation of population mobility.

Visualizations of large trajectory sets are often hardly legible because of data intersections and overlap. Andrienko and Andrienko (2010), therefore, present a method for spatial generalization and aggregation of movement data which transforms trajectories into aggregate flows between areas. These flows are subsequently used for interactive visual exploration. Their idea to transform trajectories into movements between larger geographic areas is similar to the aggregation step in our approach. However, both approaches differ in that a) we perform also a disaggregation because we still want to evaluate movement on a small geographic scale and b) the formation of the spatial aggregation units is different. While Andrienko and Andrienko (2010) determine spatial units based on a clustering of characteristic trajectory points and subsequently form spatial units as Voronoi tessellation of the resulting cluster centers, our tessellation of space is independent of the provided trajectory data. Instead, we rely on the underlying street network in order to derive spatial units that comprise micro-movements of similar type.

The aggregation of trajectories is also an important data anonymization technique. Nergiz et al. (2009) extend the notion of k-anonymity for trajectory data and Monreale et al. (2010) combine k-anonymity with the generalization approach of Andrienko and Andrienko (2010). Again, the focus of these methods is to aggregate data and additionally to conceal precise movements. Furthermore, the aggregation depends completely on the provided trajectory data and is not generic for some population.

3 Modeling Overview

In this section we give an overview of our approach to model micro-variability in mobility data. We explain the workflow and introduce all involved components. In addition, we discuss assumptions and limitations of our approach.

Our general idea to increase spatial variation of trajectory data is to separate macro- and micro- mobility. While we keep movements on the macroscopic scale, we blur movements on the microscopic level. We do this by first aggregating all trajectories to a coarser level of spatial granularity and by subsequently disaggregating the data based on traffic information. In this way we preserve the basic shape of movements, however, we stabilize and vary particular movements of the data sample with additional

background knowledge. The blurring is achieved by repeated simulation. Each simulated world contains sharp movements on the street level. However, the combination of all worlds results in a traffic distribution which provides positive visiting probabilities for all street segments under consideration.

Figure 1 shows the components and the workflow of our approach. The first component is the trajectory data sample with its mapping to some street network. Second, we need a system of spatial units or more specifically a partition of the area of interest, which can be used to aggregate the trajectory data. We call this partition *mobility unit system*. The last component of our approach is a map with traffic frequencies, which states how many people pass segments of the street network within a given amount of time.

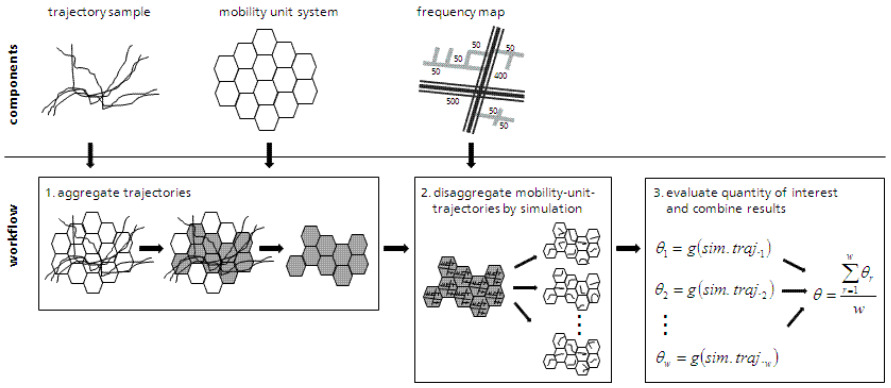


Fig. 1. Components and workflow for the increase of spatial variance of a trajectory data sample

Our approach uses the different components as follows. In the first step, the spatial granularity of the trajectories is coarsened by transforming sequences of street segments into sequences of spatial units. For this purpose, the trajectories are intersected with the mobility unit system. In the second step, we disaggregate the resulting trajectories in a repeated simulation. In each simulation, we substitute each mobility unit of the trajectories with a set of street segments that are drawn according to the traffic distribution within the unit. Of course, each simulation by itself is as unstable as the original trajectories. However, their combination yields a spatial distribution of movement on the microscopic level. Finally, the parameters of interest are evaluated for each simulation and combined.

Our approach aims at the realistic presentation of spatial variability in a mobility data set. We hereby assume a discretized geographic space where

movement takes place on the street network only. Further, our approach concentrates on the spatial distribution of movement and does not preserve characteristics as, for example, speed or direction of trajectory parts. In addition, the trajectories may lose their connectivity during disaggregation.

4 Aggregation of Trajectories

In this section, we describe the aggregation of trajectory data utilizing a system of *mobility units*. We begin with a number of considerations that have a strong impact on our algorithm and the form of the mobility unit system, followed by the introduction of the mobility unit system. Third, we describe our aggregation algorithm which utilizes the system of mobility units to merge trajectory information.

4.1 Modeling Considerations

Depending on the shape and size of spatial aggregation areas, different characteristics of mobility data are preserved. As the main goal of our approach is to increase micro-variability, we aim to form units with homogeneous mobility behavior. More precisely, we want to

a) preserve the homogeneity and variety of the underlying mobility,
b) preserve macro-mobility,
c) size the resulting areas according to the need of mobility information,
d) divide mobility according to transitional and local movement characteristics,
e) restrict mobility according to natural and artificial barriers, and
f) ensure no areal gaps and no overlaps of the mobility unit system.

The first consideration (a) refers to a similar characteristic of all mobility inside a mobility unit. For example, a downtown shopping district is characterized by a large amount of pedestrian movement while the surrounding traffic ring carries mostly vehicles. In order to preserve these movement structures, we need to form mobility units with homogeneous movement characteristics inside of them. Preserving macro-mobility (b) means that movements should keep their characteristic form on the macroscopic level. For example, a person moving in the south of a city should not be distributed to the north. The size of each unit (c) should not be too

small as this results in less variety and the continued problem of sparseness. On the other hand, oversizing dilutes the mobility information. In general, mobility units can be smaller the higher the traffic load is that they carry because it is then more likely to capture movement information in the data sample. Furthermore, we want to keep the separation of *transit* and *local* movement (d). If a person moves on a main road, we do not want to distribute his/her movement into the neighboring living area. The consideration of barriers (e) reflects the fact that mobility is street-bound and people cannot walk over water or through walls. The last consideration is a modeling requirement and states that the tessellation of the plane shall be complete and each street segment is assigned to one and only one mobility unit.

4.2 Tessellating the Mobility Space

We construct our mobility units by spatial partitioning respectively splitting the plane into a number of regions based on the street network. Clearly, the street network allows inferring movement characteristics only to a certain limit. However, it is a widely-available data source of high quality and therefore allows applying our approach in different countries. Our main idea is to split regions by transitional traffic flows. These street segments form so-called *split lines* and partition (together with natural barriers) the geographic space into *mobility units*. As the generation of the mobility unit system is a complex process, we have restricted our description to major steps. Further details and answers to remaining questions may be found in a separate follow-up paper.

In literature, many other methods have been proposed for a systematic tessellation of a plane into subareas. Voronoi decomposition is a well known method for tiling space according to a given set of fixed points and some distance function (Ottmann and Widmayer 2002). For mobility studies this implies that centers of mobility need to be known in order to obtain a Voronoi diagram. A shortcoming of this method is the lack of consideration of artificial and natural barriers. Rivers, construction sites or streets are ignored. The same shortcoming applies to official regions as postal or administrative areas which both disregard mobility flows and mingle local characteristics. Our system of mobility units is designed to overcome these shortcomings.

Let $A = \{x_1, \ldots, x_n\}$ be a space or area in which we investigate mobility behavior and x_i is a coordinate in Euclidian space $\mathbb{R}^2$. The individual mobility in the area is bound to streets segments. A street network $S = (N, E)$ is a set of nodes $N \subseteq A$ and edges $E \subseteq N \times N$ (see Fig. 2(1)).

We divide the mobility space A into *mobility units* u according to characteristics of the street network. As the street network channelizes traffic and further contains valuable mobility information, it offers a good basis for subdivision according to mobility characteristics. We differentiate two types of mobility units: *border units* u^b are built of street segments primarily carrying commuter or transitional traffic, whereas the enclosed areas are labeled *inner units* u^i. The result of the tessellation is the *mobility unit system* U with $U = \{\{u^b\}, \{u^i\}\}$. The process of constructing the unit system comprises three sequential steps (see Figure 2):

1: identify superordinate street segments (split segments),
2: split area into mobility units (regions),
3: construct mobility units.

In step 1, we define two classes of street segments $C = (C_1, C_2)$ which we call "superordinate class" (C_1) and "subordinate class" (C_2). Elements of the superordinate class $S_{C_1} \subseteq S$ primarily show a transit and commuter mobility, whereas segments of the subordinate class $S_{C_2} \subseteq S$ show mainly local mobility. We assign each street segment to one of the two classes based on traffic characteristics which we take from the street network according to a function $h: (S) \rightarrow C$. This approach has the advantage that we have complete information for the whole street network. However, if other mobility information is available, for example, traffic loads or traffic flows, they can also be used for classification. A sample result of the classification is shown in Figure 2(2). Members of class C_1 (double lines in Figure 2(2)) are also named *split segments* as they are used in step 2 to define split lines tiling the space into subareas.

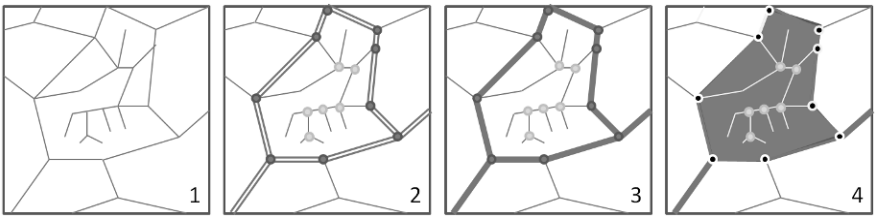


Fig. 2. 3-step-process of constructing mobility units based on the street network; (1) street network, (2) identification of superordinate street segments, (3) building of split circuits (closed paths) to tile the space and (4) construction of mobility regions

In step 2, we create a set of closed paths (see Figure 2(3)) based on all superordinate segments $\hat{s} \in S_{C_2} = (N_{C_2}, E_{C_2})$. A closed path or *split line* l is defined as a list of nodes of all street segments classified as $C_1$where the first and the last node in l are identical and no segment in the list is passed twice, more formally $l = (n_1, \ldots, n_k, n_1) \mid \forall n_j, n_{j+1} \exists \hat{s} = (\overline{n_j n_{j+1}})$ and $n_j, n_{j+1} \in N_{C_2}$. Overlaps or self-crossings are not allowed. The total set of split lines is denoted by L. The split lines are used to disjoin the street network on each split node $n \in N_{C_2}$ (see Figure 2(3)). The enclosed area of each closed path forms an *inner unit* u^i whereas the split lines form the basis for the later *border unit* u^b. All mobility units can be represented either as point set in $\mathbb{R}^2$ or as set of street segments. More formally, the former representation of mobility unit u, also called *mobility unit region*, is defined as $A_u = \{x \mid x \in A, h_A(u, x) = true\}$ where the function $h_A: (U, A) \rightarrow \{true, false\}$ tests whether a point $x \in A$ belongs to a mobility unit or not (see Figure 2(4)). This test can be performed, for example using the function *enclosure*. The latter representation describes a mobility unit u as set of street segments and is defined as $S_u = \{x \mid x \in A, h_S(u, x) = true\}$ with $h_S: (U, S) \rightarrow \{true, false\}$ testing, for example, whether the midpoint or center of gravity of a street segment lies inside A_u. In addition to the above described split lines, we also use the geometries of natural barriers as, for example, rivers or railroads to split mobility units.

The resulting mobility unit system is complete as the union of all units results in the mobility area, i.e. $\cup A_u = A, \cup S_u = S \mid u \in U$. Every street segment belongs to exactly one mobility unit.

In our case study we use the NAVTEQ street network and group each segment into classes C_1 or C_2 according to their functional class. This label discriminates street segments according to their speed, volume, and official declaration. For class C_1 we selected all segments with a functional class in {1, 2, 3} and for C_2 we selected segments with a functional class in {4, 5}.

4.3 Trajectory Aggregation

We aggregate trajectory-based mobility information using our system of mobility units. Each trajectory is transformed from an ordered list of street segments into a sequence of mobility units.

We denote a trajectory on street level by $t^S = (s_1, s_2, \ldots, s_k)$ with $s_i \in S, i = (1, 2, \ldots, k)$ and k the length of the trajectory. Similarly, we denote a trajectory on the level of mobility units by $t^U = (u_1, u_2, \ldots, u_m)$ with $u_i \in U, i = (1, 2, \ldots, m)$ and m the length of the trajectory. The entire

sets of all trajectories are denoted by T^S and T^U. The transformation possesses thus the following formalization:

$$t^S = (s_1, \ldots, s_k) \xrightarrow{\text{GenMobility}()} t^U = (u_1, \ldots, u_m), \quad s_i \in S, u_i \in U, m \leq k.$$

Alg. 1: GenMobility - generalization of mobility information

Input:
= mobility unit system U,
= $S_{u_j} \; \forall u_j \in U$ are the sets of street segments of mobility units u_j,
= street network S,
= trajectory sample data T^S on street network

Output:
= $\mathbf{T}^U$, set of trajectories where each trajectory is a list of mobility units $t_i^U = (u_1, \ldots, u_m)$

Method:

```
1:  T^U = {}
2:  t^U = ()
3:  previousUnit ← ∅
4:  for each trajectory t^S ∈ T^S do
5:      for each street segment s_i of trajectory t^S = (s_1, s_2, .., s_k) do
6:          for each mobility unit u_j ∈ U do
                # if trajectory segment inside mobility unit
7:              if s_i ∈ S_uj then
                    # checks whether the person is already
                    #  inside the unit, re-entries allowed
8:                  if previousUnit = ∅ or previousUnit ≠ u_j then
9:                      t^U = append(t^U, u_j)
10:                     previousUnit = u_j
11:                 end if
12:             end if
13:         end loop
14:     end loop
15:     T^U = insert(T^U, t^U)
16: end loop
```

Algorithm 1 shows the trajectory transformation. For each single segment we search for the mobility unit this segment belongs to. Because each segment is assigned to one unit only we attain an ordered list of consecutive mobility units according to the sequence of segments the person

passed. Each contact with one unit is stored only once except the sequence is interrupted by another unit: $(u_1, u_2, u_2, u_1) \Rightarrow (u_1, u_2, u_1)$ or $(u_1, u_1, u_2, u_3) \Rightarrow (u_1, u_2, u_3)$. The result of our generalization algorithm is a set of trajectories being mapped to lists of mobility units T^U. The generalized trajectories contain movement information on the macroscopic level. By their aggregation, they counteract sparseness of mobility data; however, all microscopic movements within a unit are lost. Our solution to this problem is a disaggregation step, which is described in the next section.

5 Disaggregation of Trajectories

In this section, we describe the disaggregation step of our approach. We start with a description of the frequency map component.

5.1 Frequency Map

May et al. (2008a, 2008b) developed a frequency map that states the average number of people that travel on a street segment by car, on foot, or by public transportation per hour for all street segments in Germany. Figure 3 shows the frequency map of the city center of Cologne, Germany. Dark colors correspond to high frequencies, light colors to low frequencies.

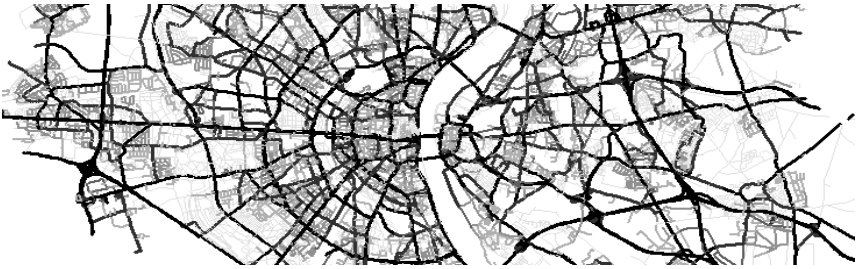


Fig. 3. Frequency map of city center of Cologne for car

May et al. (2008a, 2008b) build the frequency map based on a spatial k-nearest neighbor algorithm (*s-kNN*). The algorithm relies on a set of sample traffic frequencies and predicts values for all other street segments using geographic neighborhood relationships as well as demographic, socio-economic and POI background knowledge. For a given street network

$S = \{s_1, s_2, \ldots, s_{|S|}\}$, we will denote the corresponding frequency map with $F = \{f_1, f_2, \ldots, f_{|S|}\}$, where f_i states the frequency of segment s_i and $|\cdot|$ denotes the size of a given set. Note that we reduce the graph representation of the street network to a set of street segments in this section.

5.2 Trajectory Disaggregation and Variable Evaluation

In the disaggregation step, we map the generalized trajectories back to the street network. The mapping process is repeated several times so that a number of "trajectory worlds" are created. The trajectory information of each world is evaluated separately and afterwards the results are combined.

Alg. 2. Transformation of frequency map into traffic distribution

Input:

= street network $S = \{s_1, s_2, \ldots, s_{|S|}\}$,

= frequency map $F = \{f_1, f_2, \ldots, f_{|S|}\}$,

= mobility unit system $U = \{u_1, u_2, \ldots, u_{|U|}\}$, each mobility unit consists of a set $S_{u_i} = \{s_{t_1}, s_{t_2}, \ldots, s_{t|S_{u_i}|}\} \subseteq S$ of street segments

Output:

= set of traffic distributions $D = \{d_{u_1}, d_{u_2}, \ldots, d_{u|U|}\}$ with a probability distribution d_{u_i} for each mobility unit u_i

1: **for all** $u_i \in U$ **do**

2: **for all** $s_{t_j} \in S_{u_i}$ **do**

3: calculate $pr(s_{t_j}) = f_{t_j} / \sum_{p=1}^{|S_{u_i}|} f_{t_p}$

4: **end for**

5: **end for**

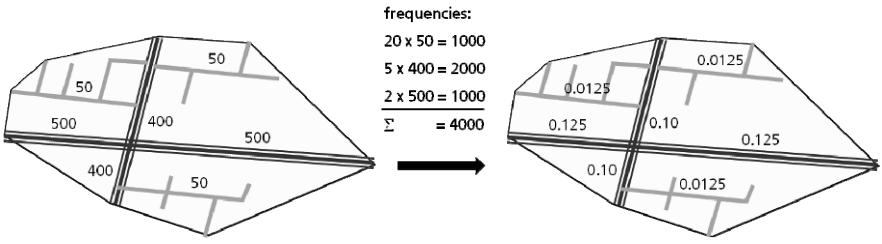


Fig. 4. Transformation of frequency map into traffic distribution for a given mobility unit

The disaggregation of a trajectory is performed per mobility unit according to the traffic distribution given by the frequency map. Therefore, our first step is to transform the frequency map into a traffic distribution for each mobility unit. The algorithm for the transformation is depicted in Algorithm 2, and Figure 4 illustrates this process. Basically, the resulting distributions describe the instantaneous probability that a person resides on a given street segment if the person is inside a given mobility unit.

Next, we need to determine with how many street segments we will substitute the passage of a mobility unit. We derive this information from the original trajectory data by calculating the average number of passed street segments of all trajectories that pass a given mobility unit. More formally, for a given mobility unit u_i with street segments $S_{u_i} = \{s_{i_1}, s_{i_2}, \ldots, s_{i|S_{u_i}|}\}$, trajectory sets T^S and T^U, $t_j^S = (s_{j_1}, s_{j_2}, \ldots, s_{j_k})$ and $t_j^U = (u_{j_1}, u_{j_2}, \ldots, u_{j_m})$ with $j = 1..|T^S|$, the average number of passed street segments in mobility unit u_i is calculated as

$$n_{u_i} = \frac{\sum_{t_j^S \in T^S} \sum_{s_{j_p} \in t_j^S} I\left(s_{j_p} \in S_{u_i}\right)}{\sum_{t_j^U \in T^U} \sum_{u_{j_q} \in t_j^U} I\left(u_{j_q} = u_i\right)} . \tag{5.1}$$

In the above equation, $I(\cdot)$ denotes a Boolean function which has a value of 1 if its argument is true and 0 if its argument is false. Note that instead of using the average number of passed street segments, it would also be possible to sample a number from the distribution of passed segments for each mobility unit passage. For simplicity, however, we chose the average.

Knowing the traffic distribution and the average number of segment passages for each mobility unit, we can disaggregate a unit passage of u_i by n_{u_i}-times random sampling from the mobility unit's traffic distribution. The disaggregation of all mobility-unit-trajectories yields one simulated trajectory world $\dot{T}^S$, on which the quantity of interest can be evaluated. As we selected n_{u_i} as the average number of passed segments per mobility unit, the simulated trajectory world contains the same number of street segments as the original set of trajectories. If we repeat the simulation process for a large number of times, the distribution of all selected street segments will converge to the traffic distribution of the mobility unit due to the law of large numbers. Algorithm 3 shows the disaggregation process. Hereby, function *multinomial* denotes the random sampling from a

multinomial distribution. The distribution is given in the first argument and the number of samples drawn in the second.

Alg. 3. Disaggregation of mobility-unit-trajectories based on repeated simulation

Input:

= mobility-unit-trajectory sample T^U,

= set of traffic distributions $D = \{d_{u_1}, d_{u_2}, \ldots, d_{u_{|U|}}\}$,

= set of the average number of passed street segments per mobility unit $= \{n_{u_1}, n_{u_2}, \ldots, n_{u_{|U|}}\}$,

= number of simulations w

Output:

= set of simulated trajectory words on street level $TW = \{\dot{T}_1^S, \dot{T}_2^S, \ldots, \dot{T}_w^S\}$

```
TW = {}                                  # initialize set of trajectory worlds
for r = 1..w do                          # perform w simulations
    Ṫ_r^S = {}                           # initialize single trajectory world
    for all t_j^U ∈ T^U do               # for each aggregated trajectory
        t_j^S = ()                       # initialize simulated trajectory
        for all u_{j_i} ∈ t_j^U do       # for each mob. unit per trajectory
            # randomly draw n_{u_{j_i}} segments and append to trajectory
            t_j^S = append(t_j^S, multinomial(d_{u_{j_i}}, n_{u_{j_i}}))
        end for
        Ṫ_r^S = insert(Ṫ_r^S, t_j^S)    # insert trajectory to trajectory set
    end for
    TW = insert(TW, Ṫ_r^S)               # insert simulation to world set
end for
```

For each generated trajectory world, the quantity of interest can be calculated by some function g and the results can be combined e.g. by averaging over all trajectory worlds, i.e.

$$\theta_r = g\left(\dot{T}_r^S\right) \qquad \forall r = 1 \ldots w, \tag{5.2}$$

$$\theta = \frac{\sum_{r=1}^{w} \theta_r}{w}. \tag{5.3}$$

Due to random sampling, the simulated trajectories lose their connectivity within the mobility units. We can thus evaluate only quantities that are (for the most part) independent of the connectivity property. In future work, we plan to improve the disaggregation process by sampling short trajectories instead of a set of street segments. However, this makes the sampling process more complex as it means generating a trajectory set per mobility unit that conforms to the traffic distribution and to the number of passed street segments.

6 Application Scenario

6.1 Outdoor Advertising

In 2007 the Arbeitsgemeinschaft Mediaanalyse (ag.ma), the governing organization of German advertising, commissioned a first study to evaluate poster performance based on mobility data. The study includes a mobility survey for a measurement period of 7 days using GPS and CATI technology. This technology has the advantage that poster performance can be differentiated with respect to the location of poster sites as well as the sociodemography and origin of target groups. Performance measures of interest for the advertising branch are gross rating points (GRP) and reach. GRP specify the total number of poster contacts that 100 persons of a respective population produce with a given campaign in a given period of time. Reach describes the percentage of the population that passes at least one poster of the campaign within a given period of time.

Today, the survey includes 42,780 participants for up to 7 measurement days. However, the trajectory data sample poses the challenge that not all poster locations are visited and therefore performance measures cannot be calculated with conventional trajectory analysis. Figure 5 (left) visualizes this exemplarily. If a location without GPS trajectories is evaluated, all performance measures will be zero. This, however, is not the true performance value but results from missing variability in the data sample.

In order to calculate equitable performance measures it is necessary to react to the lack of empirical variance. This can be done by applying our approach to increase spatial micro-variability. As shown in Figure 5 (right) each segment of the mobility unit receives a positive selection probability within the simulations.

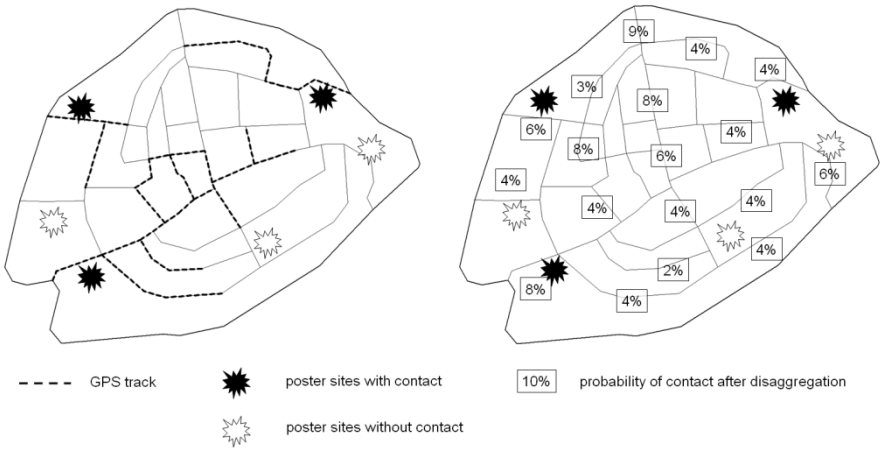


Fig. 5. Example of poster contacts in a mobility unit

We evaluated our approach by comparing performance measures calculated once on the given sample trajectories and once on the simulated trajectories for differently sized campaigns. The results showed a high similarity for all cases.

6.2 Mobility Unit System by Example

For our application we generated a system of mobility units for all of Germany, which contains 194,331 units. In the following, we show example statistics and pictures for the city of Cologne. Cologne comprises in total 2,256 mobility units (see Figure 6).

As Table 1 shows, most of the units are border units. They are comparably small; however their size is sufficient due to the high traffic volume on these streets.

Table 1. Statistics of mobility unit system for Cologne, Germany

	Number of mobility units	Average number (and median) of street segments per unit	Average number (and median) of passed street segments (n) per unit
border units	1,997	4.6 (1.0)	2.4 (1.0)
inner units	259	137.9 (43.0)	5.4 (4.8)
Total	2,256	19.9 (1.0)	2.8 (1.1)

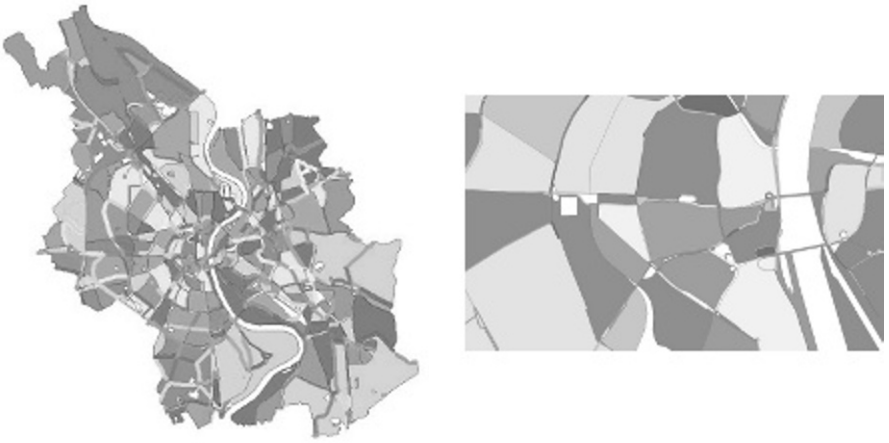


Fig. 6. Mobility unit system Cologne (whole city and magnification of city center)

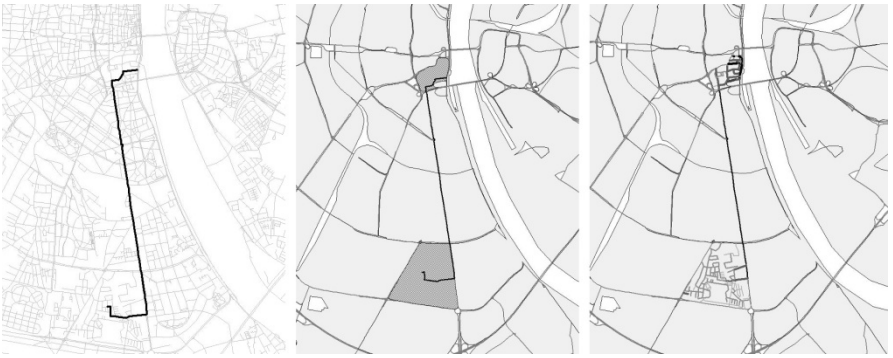


Fig. 7. Original, aggregated and simulated example trajectory in Cologne

Figure 7 shows the aggregation and disaggregation of an example trajectory in Cologne. The trajectory starts and ends in an inner unit, which allows varying the trajectory in the respective areas during simulation. In between, the trajectory passes only border units where the movement variation is small. This distinction between border and inner units is reasonable because border units carry more traffic and therefore need less variation. In addition, it keeps the movement compact and avoids mixing transitional and local traffic. On the macroscopic level, the movement information is preserved.

7 Summary and Future Research Challenges

A growing number of companies and research institutions are interested in the analysis of mobility data with demand of a high level of spatial detail. However, existing mobility studies cannot comply with this demand, because they do not capture all spatial variation of the population. We therefore present an approach to increase the spatial variability of a given mobility data set while retaining important mobility characteristics of the sample. Our approach consists of two steps. In the first step, we use spatial aggregation to generalize movements on a coarse level of spatial detail. This ensures robust sample sizes as well as the preservation of movement characteristics on the macroscopic scale. In the second step, we disaggregate the data based on knowledge about the traffic distribution using repeated simulation. In this way, we introduce spatial variability to the mobility data on a fine level of resolution while stabilizing the disaggregation.

Our approach is implemented in a real-world business application for the German outdoor advertising industry. It constitutes an important model part for the evaluation of poster performance. This evaluation is a business-critical task in outdoor advertisement and has been successfully applied since 2007 (ag.ma 2010).

In future work, we intend to improve the disaggregation step by sampling trajectories instead of street segments for the passage of a mobility unit. In this way, the simulated trajectories become sequences of short trajectories which are connected within each mobility unit. In addition, we plan to adapt our approach for the analysis of mobile phone data. Differing from our current proceeding, the mobility units of these data are already given by the layout of the mobile cells. As these cells do not follow particular mobility criteria, it will be a challenge to preserve typical mobility characteristics. In addition, the geometries of mobile cells can overlap, which introduces additional complexity.

References

ag.ma (2010) Arbeitsgemeinschaft Media-Analyse e.V. (German working group for media analysis) http://www.agma-mmc.de, last date accessed Jan 2010

Adrienko N, Adrienko G (2010) Spatial generalization and aggregation of massive movement data. IEEE Transactions on Visualization and Computer Graphics. IEEE Computer Society

BMVBS – Bundesministerium für Verkehr, Bau und Stadtentwicklung (2010) Mobilität in Deutschland 2008, Abschlussbericht (Mobility in Germany 2008, final report). http://www.mobilitaet-in-deutschland.de

DDS – Digital Data Services GmbH (2010) Arbeitswegematrix (commute matrix for Germany). http://www.ddsgeo.com/products/arbeitswegematrix.html

FAW – Fachverband Außenwerbung e.V. (2010). Netto-Werbeeinnahmen erfassbarer Werbeträger in Deutschland, 2000-2009 (Net turnover of confirmable advertising media in Germany 2000-2009). http://www.faw-ev.de/media/
download/marktdaten/4_Nettoumsaetze_aller_Werbemedien_ab_2001.pdf

Giannotti F, Nanni M, Pedreschi D, Pinelli F (2007) Trajectory pattern mining. In: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07). ACM, pp 330-339

Gudmundsson J, Kreveld M, Speckmann B (2007) Efficient detection of patterns in 2D trajectories of moving points. In: Geoinformatica 11(2):195-215

Laube P, Imfeld S (2002) Analyzing relative motion within groups of trackable moving point objects. In: Proc. of the 2nd International Conference on Geographic Information Science (GIScience'02). Springer, pp 132–144

May M, Hecker H, Körner C, Scheider S, Schulz D (2008a). A vector-geometry based spatial knn-algorithm for traffic frequency predictions. In Proc. of the 2008 IEEE International Conference on Data Mining Workshops (ICDMW '08). IEEE Computer Society, pp 442-447

May M, Scheider S, Rösler R, Schulz D, Hecker D (2008b). Pedestrian flow prediction in extensive road networks using biased observational data. In Proc. of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS '08). ACM, pp 1-4

Monreale A, Andrienko G, Andrienko N, Giannotti F, Pedreschi D, Rinzivillo S, Wrobel S (2010) Movement data anonymity through generalization. Transactions on Data Privacy 3(2):91-121

Nanni M, Pedreschi D (2006) Time-focused density-based clustering of trajectories of moving objects. In: Journal of Intelligent Information Systems (JIIS), 27(3):267-289, Special Issue on Mining Spatio-Temporal Data

Nergiz ME, Atzori M, Saygin Y, Guc B (2009) Towards trajectory anonymization: a generalization-based approach. Transactions on Data Privacy 2 (1)

Ottmann T and Widmayer P (2002) Algorithmen und Datenstrukturen. Spektrum, Lehrbuch, Volume 4, Heidelberg Berlin

Pelekis N, Kopanakis I, Ntoutsi I, Marketos G, Andrienko G, Theodoridis Y (2007) Similarity search in trajectory databases, In: Proc. of the 14th IEEE In-

ternational Symposium on Temporal Representation and Reasoning (TIME 2007). IEEE Computer Society Press, pp 129-140

Rinzivillo S, Pedreschi D, Nanni M, Giannotti F, Andrienko N, Andrienko G (2008) Visually driven analysis of movement data by progressive clustering. In: Information Visualization 7(3):225-239

Wolf J (2003) Tracing people and cars with GPS and diaries: Current experiences and tools. Presentation at ETH Zurich. http://www.ivt.ethz.ch/vpl/publications/presentations/v53.pdf

Wolf J, Guensler R, Bachman W (2001) Elimination of the travel diary: an experiment to derive trip purpose from GPS travel data. Transportation Research Board 80th Annual Meeting

Yang Y, Hu M (2006) TrajPattern: mining sequential patterns from imprecise trajectories of mobile objects. In: Proc. of 10th International Conference on Extending Database Technology. Springer, pp 664-681

Zheng Y, Zhang L, Xie X, Ma WY (2009) Mining Interesting locations and travel sequences from GPS Trajectories. In: Proc. of the 18th International World Wide Web Conference (WWW'09). ACM, pp 791-800

GeoSensor Development and Application

The SID Creator: A Visual Approach for Integrating Sensors with the Sensor Web

Arne Bröring[1,2,3], Felix Bache[2,3], Thomas Bartoschek[2], Corné P.J.M. van Elzakker[1]

[1]Faculty ITC, University of Twente, Netherlands
[2]Institute for Geoinformatics, University of Muenster, Germany,
[3]52°North Initiative for Geospatial Open Source Software, Germany
broering@52north.org, {felix.bache,bartoschek}@uni-muenster.de,
elzakker@itc.nl

Abstract. This paper describes the *Sensor Interface Descriptor* (SID) model and focuses on presenting and evaluating the *SID creator*, a visual approach to create instances of the SID model. Those SID instances comprise the knowledge required to integrate a sensor with the Sensor Web. This integration is done by an *SID interpreter* which uses an SID instance to translate between a sensor protocol and the Sensor Web protocols. An SID instance, designed for a particular sensor type, can be reused in multiple applications and can be shared among user communities. The SID creator enables users to describe the interface, commands and metadata of their sensors. In a user study, we evaluated the simplification of the sensor integration process through the SID concept. The study incorporated four user groups, ranging from high school students to expert users, who were challenged to integrate weather station sensors with the Sensor Web by utilizing the SID creator. While the common approaches of integrating such sensors with the Sensor Web involve manual coding and extensive adaptation efforts, this new visual approach significantly simplifies the integration process.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_8,

1 Introduction

The aim of the Sensor Web is to enable discovery, access, exchange, and processing of sensor data, sensor metadata, and sensor task planning across different applications. The Sensor Web Enablement (SWE) initiative of the Open Geospatial Consortium (OGC) standardizes Web Service interfaces and data encodings which can be utilized to build such a Sensor Web [1]. The interoperable SWE services make sensors available by hiding the sensor communication details and the heterogeneous sensor protocols from applications.

In recent years, the SWE standards have been applied in various projects (e.g. [2, 3]) showing their practicability and suitability in real world scenarios. However, a central challenge still remains to be tackled. Since SWE services are defined from an *application-oriented* and not from a *sensor-oriented* perspective, the integration of sensors and services is not sufficiently defined, yet. In fact, a gap of interoperability between SWE services and sensors has been identified [4].

By looking at two SWE services (Section 2), the Sensor Observation Service (SOS) and Sensor Planning Service (SPS), this interoperability gap becomes clear. The SOS offers operations for registering sensors and uploading their data. Due to limitations in bandwidth and processing power, sensors are usually not able to transform their measured data to the SWE protocols and to upload them to the SOS. The SPS offers operations for an interoperable tasking of sensors. However, it is not defined by the specifications how an SPS server transforms a retrieved sensor task to a command of the sensor protocol.

Today, sensors are integrated with the Sensor Web by manually building proprietary bridges for each pair of SWE service implementation and sensor type. This approach is cumbersome and leads to extensive adaptation efforts - the key cost factor in developing large-scale sensor network systems [5]. Relevant concepts which facilitate the sensor integration have not been realized yet.

Minimizing those sensor integration efforts can significantly support applications such as disaster management where an ad-hoc densification of an existing sensor network is demanded. Examples range from flooding scenarios, in which the affected river courses are not covered densely enough with water gauges, to incidents in nuclear plants, which require ad-hoc deployments of radiation detectors. Assuming a Sensor Web is already in place and used by disaster relief organizations as a coherent infrastructure to access sensors, an integration of new sensors in the most efficient way becomes necessary.

This work focuses on presenting and evaluating a visual creator for Sensor Interface Descriptors (SID). This SID creator facilitates the integration of sensors with the Sensor Web by enabling a semi-automatic generation of their interface and protocol descriptions. After describing the SWE initiative and related work (Section 2), an overview of the SID model[1] is presented (Section 3). It enables the declarative description of a sensor's interface. Based on this model, SID interpreters can be built which use the knowledge contained in an instance of the SID model to translate between sensor protocol and Sensor Web protocols. Such interpreters for SID instances can be built independently of particular sensor technology. Hence, the SID model, together with generic SID interpreters, closes the interoperability gap between sensors and the Sensor Web as described above.

However, the manual creation of SID instances is not straightforward. Therefore, the visual SID creator has been developed (Section 4) which supports users in describing the sensor interface to integrate the sensor with the Sensor Web. We evaluated this SID creator and the SID concept by conducting a user study (Section 5). The participants of the user study, nine senior high school students and eleven people with at least a Bachelor of Science degree in Computer Sciences were challenged to integrate the sensors of a weather station with the Sensor Web by utilizing the SID creator. In Section 6, the results of the user study are analyzed. The paper ends with a conclusion and gives an outlook to future work.

2 Background & Related Work

The main Web Services of OGC's SWE framework are the Sensor Observation Service (SOS) and the Sensor Planning Service (SPS). The SOS [6] provides interoperable access to sensor data as well as sensor metadata. To control and task sensors the SPS [7] can be used. A common application of SPS is to define simple sensor parameters such as the sampling rate but also more complex tasks such as mission planning of satellite systems.

Apart from these Web Service specifications, SWE incorporates information models for observed sensor data, the Observations & Measurements (O&M) [8] standard, as well as for the description of sensors, the Sensor Model Language (SensorML) [9].

SensorML specifies a model and encoding for sensor related processes such as measuring or post processing procedures. Physical as well as logical sensors are modeled as *processes*. The functional model of a process

[1] A detailed description of the SID model can be found in [18] and [19].

can be described in detail, including its identification, classification, inputs, outputs, parameters, and characteristics such as a spatial or temporal description. Processes can be composed by process chains.

O&M defines a model and encoding for *observations*. An observation has a result (e.g. 0.7 mSv/a) which is an estimated value of an *observed property* (e.g. radiation), a particular characteristic of a *feature of interest* (e.g. the city of Muenster). The result value is generated by a *procedure*, e.g. a sensor such as a radiation detector described in SensorML. These four central components are linked within SWE.

So, while the connection between the Sensor Web layer, consisting of those SWE components, and the application layer is well-defined, the connection between Sensor Web layer and sensor layer is not yet sufficiently defined. This interoperability gap between the two layers can be addressed from two directions. By following a bottom-up approach the interoperable access on the sensor layer is improved. Top-down approaches introduce mechanisms on the Sensor Web layer to abstract from the variety of sensor protocols (Figure 1).

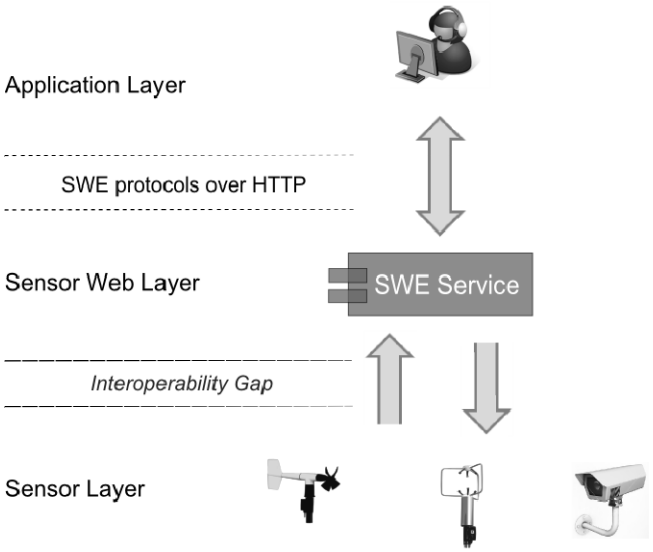


Fig. 1. The Sensor Web layer stack

The bottom-up direction is addressed by several standardization approaches. Promising is the IEEE 1451 family of standards[2] which is a universal approach to connect sensors to diverse networks and systems. An

[2] http://ieee1451.nist.gov/

important feature of this standards family is the definition of a Transducer Electronic Data Sheet (TEDS) which is a small memory device attached to the transducer describing for example its identification, calibration, correction data, measurement range, and manufacturer related information. However, the expressiveness of TEDS is limited and it cannot capture all metadata of a sensor. For example, higher level processing of sensor data cannot be described in TEDS. This requirement is addressed by SensorML. Therefore, Hu et al. [10] convert TEDS to SensorML by creating a knowledge base which maps each TEDS property to an appropriate SensorML description. It would be promising to extend this approach and to combine it with our work to automatically generate SIDs for IEEE 1451 sensors so that an SID interpreter can connect IEEE 1451 sensors on-the-fly with SWE services.

However, in today's real world applications not only IEEE 1451 but in fact a huge variety of sensor interfaces (standardized or proprietary) are utilized. Hence, different projects are approaching the interoperability gap in a top-down manner, from the upper Sensor Web layer.

The application AnySen [11] is capable of reading and interpreting data from sensor nodes by abstracting the sensor protocols and reading the sensor description from an external file. The authors do not detail but claim that AnySen allows the formatting of these sensor descriptions compliant to the SensorML standard. While AnySen supports the provision of sensor data by connecting to an SOS, other SWE services, in particular tasking of sensors through an SPS, are not supported.

Walter and Nash [4] identify the interoperability gap and analyze different system models which may lower the implementation barrier for coupling sensor systems and SWE services. The authors suggest lightweight SWE connectors which can be adapted to different raw sensor formats to convert them to SWE-based data models. They state that such SWE connectors could be implemented for a wide range of different sensor types. They come up with design approaches, but do not detail them.

The Sensor Abstraction Layer (SAL) [12] is most similar to the SID concept. SAL makes use of SensorML to describe sensor interfaces. As a library, it offers high-level functions to access sensors by hiding their specific technological details. The architecture follows a split design consisting of lightweight SAL agents running on the sensor gateways to handle the communication with the hardware and SAL clients usable by application developers to invoke specific actions on sensors managed by an agent. Mechanisms are missing for the final connection to SWE services and the integration of sensors with the Sensor Web.

None of the approaches above is leveraged by a visual component which supports the creation of a connector or adapter between sensor and

Sensor Web. Focus of this work is such a component which enables the visual creation of instances of the SID model.

3 Sensor Interface Descriptors

The architectural principle of a Sensor Web infrastructure including the usage of SIDs is shown in Figure 2. A sensor communicates with a data acquisition system in its specific sensor protocol over a transmission technology such as RS232 or Ethernet. This sensor can also act as a sensor gateway (network sink) so that other nodes of a (possibly mobile) sensor network communicate with it. The SID interpreter runs on the data acquisition system and uses SID instances for the different sensors of the sensor network to translate between the sensor specific protocol and the SWE protocols. The interpreter is responsible to register a sensor at a SWE service and to upload sensor data to an SOS. The interpreter is also responsible for the opposite communication direction and forwards tasks received by an SPS to a sensor.

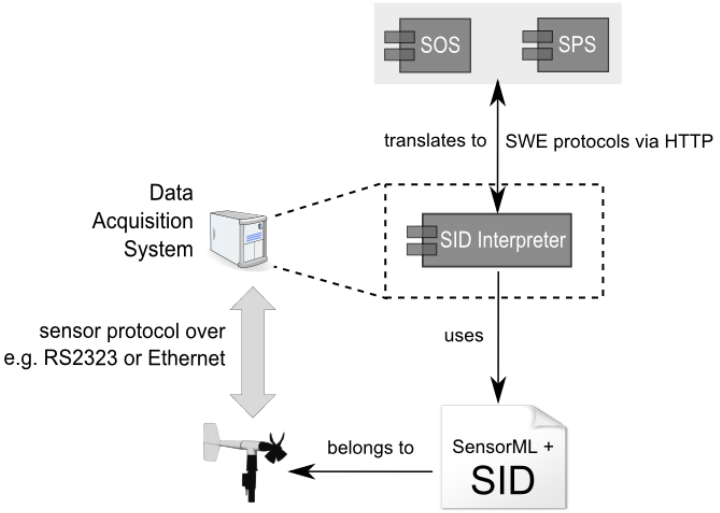


Fig. 2. Connection of a sensor to SWE services through an SID interpreter

A strong requirement of the design of the SID model is the strict encapsulation of the SID within the SensorML document. The SID part of the SensorML document is specific for a certain sensor type, not a particular sensor instance. Hence, an encapsulation allows reusing it in the SensorML descriptions of different sensors which are of the same type. The

approach developed here encapsulates the SID within the *InterfaceDefinition* element of a SensorML document.

The *InterfaceDefinition* element contains a stack of layers (Figure 3), aligned with the Open System Interconnection (OSI) reference model. In contrast to the OSI model, SensorML does not further define how to use these layers. The SID model makes use of this layer stack and concretizes its usage to describe the sensor interface.

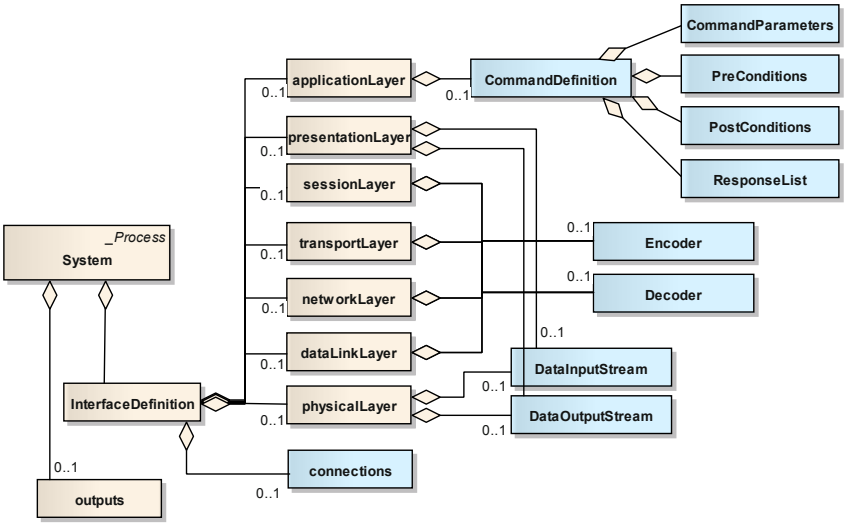


Fig. 3. Excerpt of SensorML schema (beige colored data types) and an overview of the encapsulated SID extension (blue colored data types)

The addressing parameters (e.g. port and baud rate of a serial connection) are the basis for establishing a physical connection to the sensor. This physical connection is established through the operating system which runs the SID interpreter. The addressing parameters are stored in an external document referenced by the SID, since the SID can be published publicly (e.g. via a SWE service) and the addressing parameters are security relevant.

After establishing the physical connection, a definition of the raw sensor protocol exchanged between sensor and data acquisition system is essential. We describe the structure of these raw data within the lowest, the *physicalLayer* element. As shown in Figure 3, new elements for the data input and data output stream are attached to this element. The two elements are necessary to support duplex communication with sensors.

For enabling the definition of processing steps which are necessary to translate between the sensor protocol and the SWE protocol, the *dataLinkLayer*, *networkLayer*, *transportLayer*, and *sessionLayer* are utilized. To al-

low data processing in both directions, from sensor domain to SWE domain and the other way round, elements for data decoding and encoding are added to each layer (Figure 3). Instances of these elements contain descriptions of applied processing steps. Here, the SID model reuses existing SensorML types to define processes with its inputs, outputs, parameters, and its computational method.

An example for a typical usage of the layers to process a data stream coming from a sensor and to encode it to SWE protocols can look like this: the data link layer specifies a process for character escaping, the network layer computes a checksum validation, the transport layer transforms the raw data to observations by applying an interpolation, and the session layer computes a date conversion.

The data, resulting from the preceding processing steps, have to be associated with certain metadata, which is part of the O&M model (Section 2), before it can be forwarded to an SOS. The measured data need to be associated with units of measure. Further, the data need to be linked to the elementary SWE components, the observed property and the feature of interest, so that observations of the O&M model can be built and inserted into an SOS.

While the association of the data with a unit of measure is done on the *presentationLayer*, the link to observed property and the feature of interest is established in the outputs element of the SensorML document. This outputs element is not part of the SID, since it is not a sub-element of the *InterfaceDefinition* (Figure 3). The contained information is intentionally kept out of the SID, since the linkage of a sensor to feature of interest and observed property is dependent on the particular use case, not the interface of the sensor type. By not including this information into the SID, a reusing of the SID in different SWE deployments is possible.

The application layer of the OSI model describes interfaces to access the OSI stack. Compliant to this view, the *applicationLayer* is used here to define the commands accepted by the sensor. These command definitions can be used by an SPS so that it can provide information to the clients on how to task the sensor. As shown in Figure 3, the *command* element contains sub-elements to describe possible sensor responses, the pre- and post-conditions for executing the command, as well as the command parameters.

The implementation of our SID interpreter is based on the OSGi framework[3] which is extendible by pluggable and loosely coupled components. An overview of the architectural design of the SID interpreter implementation is depicted in Figure 4.

[3] http://www.osgi.org/

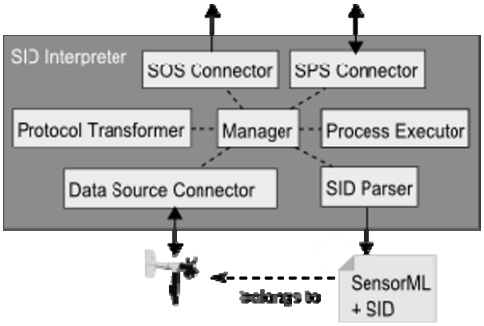


Fig. 4. Overview of SID interpreter implementation

A central *Manager* component controls the workflow. First, the *SID Parser* is used to read in the SID document of the sensor. Depending on the specified addressing parameters, a particular *Data Source Connector* implementation (e.g. for USB connections) is chosen to connect to the sensor. Based on the protocol definition of the SID, the *Protocol Transformer* communicates with the sensor in a bi-directional way. The *Process Executor* is able to execute the four native process methods. Also, user-defined MathML processes can be executed by means of the MathML Solver library[4]. The *SOS Connector* triggers the SOS operation *RegisterSensor* to add the new sensor to the Sensor Web and executes the *InsertObservation* operation to upload sensor data as observations to an SOS. The *SPS Connector* forwards the SensorML document and the contained SID to an SPS which uses the sensor command descriptions to provide detailed information on how to task the sensor. Sensor tasks, submitted to the SPS, are forwarded by the SPS to the *SPS Connector*. The tasks are transformed to the sensor protocol, and passed through the *Data Source Connector* to the sensor.

4 A Visual Creator for Sensor Interface Descriptors

The creation of SensorML and contained SID code without tool support is tedious and error-prone, since plain XML has to be written by hand. For this reason, the visual SID creator has been developed which enables a semi-automatic generation of SID instances. This SID creator follows the *wizard* user interface pattern [13] and consists (in the version used in this work) of four pages for the different aspects of the SID design. Labels and

[4] http://sourceforge.net/projects/mathmlsolver

descriptions guide the user in filling out the forms of each wizard page. Additionally, a dynamic context help can be consulted for each page which contains detailed information about all input fields. The syntactic validity of user inputs is directly checked and feedback is given in case of invalid input. The user is only able to go to the next page of the wizard if all fields are completed correctly. A bar on top indicates the overall process of the SID creation.

The first page of the wizard allows the definition of the directory where the generated SID file is saved after creation. The second page (Figure 5) prompts the user to specify basic metadata about the sensor. This includes the globally unique identification within the Sensor Web, a human readable name, and description of the sensor, as well as its geographic location. The specified data are pasted in particular SensorML tags when the file is generated. Since SensorML is generic and does not explicitly specify where to put this information, we follow a public profile of SensorML which is optimized for discovery of sensors [14] to encode these data.

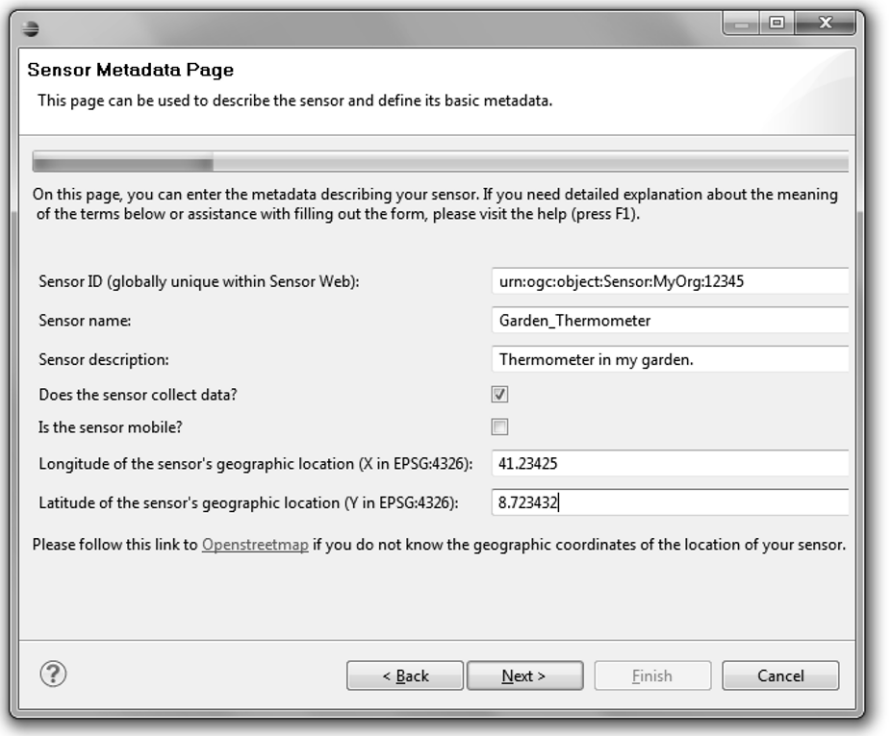


Fig. 5. Basic metadata description page of the SID creator

The third page (Figure 6) of the wizard enables the definition of the sensor protocol. First, the user chooses how the SID interpreter retrieves data from the sensor. Alternatives are, for example, the serial port, USB, Ethernet, or a file-based connection where the communication with the sensor takes place through a file on the hard disk of the data acquisition system.

Next, the separator signs of the sensor protocol are being defined. Those signs are utilized by the protocol to separate blocks, fields within a block, and decimal numbers. Afterwards, the structure of the protocol is defined. The SID creator allows specifying multiple blocks within the data stream coming from the sensor. For those blocks between 1 to n contained fields can be defined. An example of such a block is given in Listing 1 and its description with the SID creator is shown in Figure 6.

```
… # thermometer123 | 2010-09-02T13:05 | 22.34 | °C # …
```

List. 1 A single block within a sensor data stream

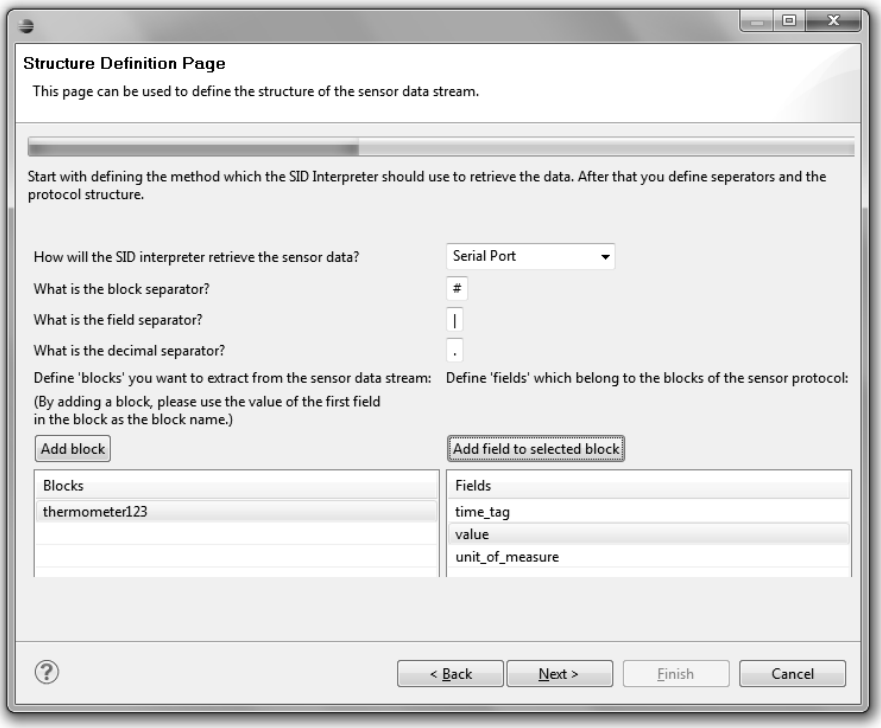


Fig. 6. Structure definition page of the SID creator

The block is identified within the sensor data stream by the value of its first field, *thermometer123* in case of Listing 1. This block ID is also specified in the wizard (Figure 6). Further, three fields are added to the block. The second field is the value of the measured data which is of interest and referenced on the next wizard page.

The fourth page (Figure 7) defines the sensor data output which shall be uploaded to the Sensor Web. Further, the SWE related metadata, such as the observation offering, feature of interest, observed property, and unit of measure (Section 2), can be associated with the sensor data output. Those metadata are needed by the SID interpreter to create O&M encoded observations and to call the *InsertObservation* operation of the SOS every time data is coming in from the sensor (in its configured sampling rate).

After finalizing this page, the wizard creates a complete SensorML description for the sensor with a contained SID as defined by the user.

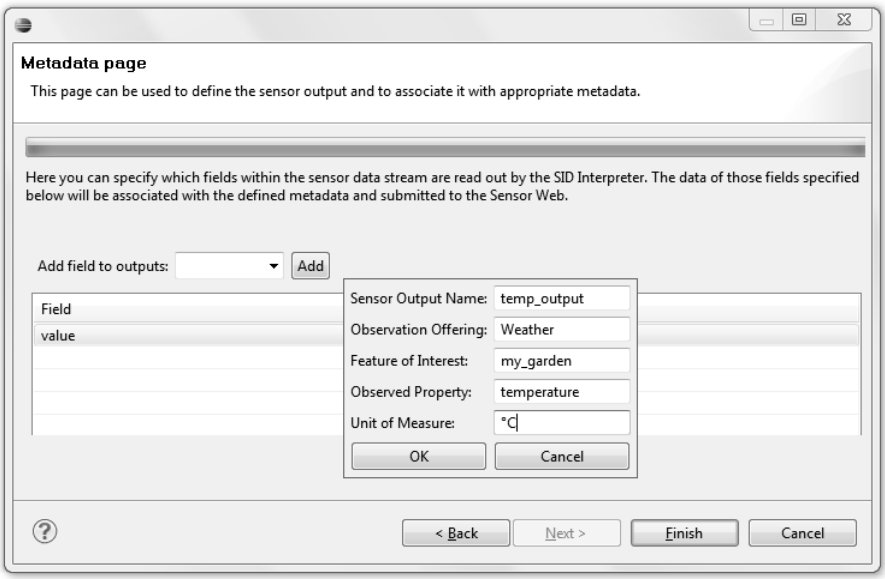


Fig. 7. Metadata association page of the SID creator

5 User Study

The user study was conducted to analyze the usability of the SID creator and to find out whether the developed concepts help to facilitate the integration of sensors with the Sensor Web. The participants were tasked to

utilize the SID creator to describe the protocol of a home weather station[5] and to associate the measured sensor data with SWE metadata. By completing the task correctly, the SID creator would output an SID file which can be used by the SID interpreter to automatically register the sensors at an SOS and translate the measured sensor data to O&M and upload it to the SOS server. This setup is depicted in Figure 8. The SID interpreter runs on a computer (data acquisition system) with a USB connected weather station. The weather station writes the measured data continuously every 10 minutes to a data file on the hard drive of the computer.

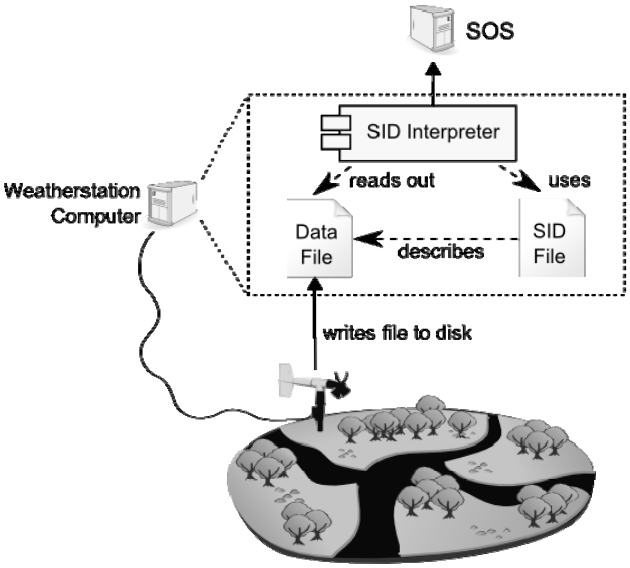


Fig. 8. Overview of the user study setup

The structure of the data file written by the weather station represents the sensor protocol which has to be described within an SID. Listing 2 shows an instance of such a data file. Besides basic metadata, the file contains measured data for a particular time stamp from a wind speed sensor, a wind direction sensor as well as a thermometer. In the user study, the participants were tasked to focus on the thermometer and to describe its protocol in an SID.

[5] A common DAVIS weather station (http://www.davisnet.com/) was chosen.

```
Sensor_Type;DavisWeatherStation#
Coordinate_System;EPSG4326#
Coordinates;52.223;7.544#
Time_Stamp;2010.09.30;12:57:46#
WindSpeedSensor;WindSpeed;34;m/s#
WindDirectionSensor;WindDirection;270;deg#
Thermometer;Temperature;22.34;°C#
```

List. 2 Weather station data file

The participants of the study were selected with the intention of having users with varying experience, so that not only the behavior of expert users, but also the behavior of users who have only little computer knowledge, could be studied. Overall, 20 people took part in the study. Nine participants were high school students[6] aged 17 to 19. Among this group, four had moderate computer experience (e.g. only office programs etc.) and no programming skills. In the following, we refer to those participants as *Group A*. The other five high school students (*Group B*) were attending a computer science course and had good computer experience and basic programming skills in Java and Delphi. Among the other eleven participants, six had at least a Bachelor of Science (BSc) degree in computer sciences (*Group C*). The other five participants (*Group D*) were the most qualified group and had at least a BSc degree in computer sciences and also experience with the SWE specification framework and the SOS in particular. None of the participants had prior knowledge of the SID concept.

All participants were given a 25 minutes introductory presentation explaining the basic idea of the Sensor Web and the relevant standards, i.e. the principle of the SOS as well as the central metadata components of SensorML and O&M (Section 2). A description of the SID concept and the weather station protocol was also part of the presentation. Due to the different levels of user experience, the presentation[7] was kept simple and did not go into encoding details. After this introduction, each participant was given a short written task description and could ask final questions to make sure the task was understood. The test was conducted by applying screen logging in combination with the "Think Aloud" method [15, 16, 17] i.e., the participants were supposed to talk about what they were doing and thinking what difficulties they had while utilizing the SID creator. The

[6] The high school students took part in a one week school project which aimed at making the school's weather station available on the Sensor Web.

[7] The interested reader can download the presentation here: http://ifgi.uni-muenster.de/~arneb/SID_Creator.pdf. Please be aware that the participants were all German native speakers, hence, the presentation as well as the text of the SID creator pages were kept in German.

voice, the screen, and the duration of each test, were recorded. During the test, the interaction between experimenter and participant was minimized. If advice or help was given by the experimenter it was taken note of and such interferences are reflected in the evaluation of the study (Section 6). After finishing the test, the participants were also asked to complete a questionnaire.

6 Analysis and Evaluation of the User Study

As expected, the experienced Group D was most successful in creating valid and working SID instances. Four of five members produced a working SID for the weather station. Two of the six participants with a BSc degree in computer science but without SWE knowledge (Group C) were also successful. In each of the two groups of high school students (Group A and B) one person created a working SID.

From analyzing the user study recordings, it is noticeable that mistakes made by the participants happened repeatedly and can be classified. Overall, the 20 participants made 45 mistakes. Thereby, it has also been counted as a mistake if a participant requested advice for a particular problem and the experimenter interfered. Figure 9 shows the average number of such mistakes per person, separated for each participant group and wizard page[8]. The diagram shows that the average number of mistakes per person decreases with increasing level of experience. The high school students (Group A and B) as well as Group C made most of their mistakes on page 3, where the structure of the sensor protocol needs to be defined. On page 4 (Figure 7), each group made almost the same number of around 0.5 mistakes per person. On page 2, very few mistakes were made (only two mistakes by members of Group B and C) showing that this page is rather easy to complete.

[8] Since no participant made a mistake on page 1 of the wizard, it is not considered here.

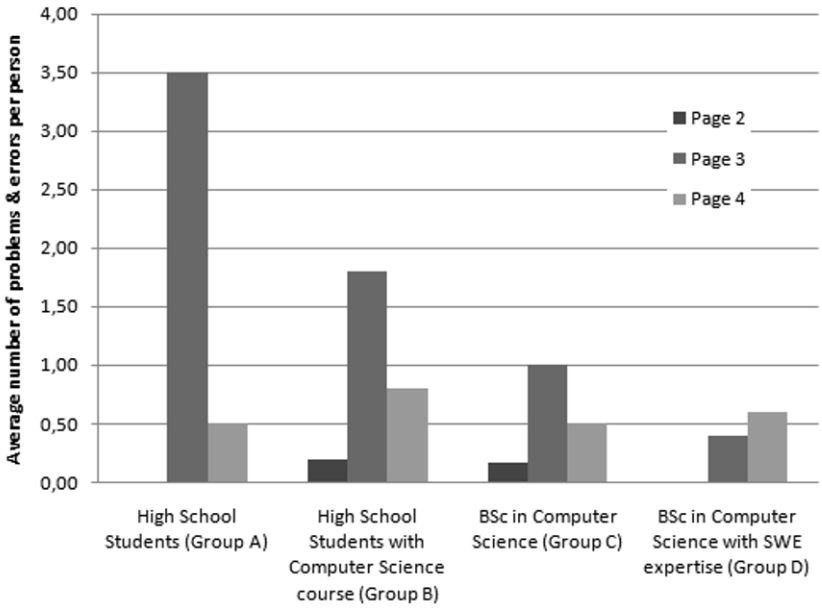


Fig. 9. Average number of mistakes per person for each wizard page

Figure 10 shows the relative frequency of the kinds of errors that occurred. Most often, namely 20 percent of all mistakes, a wrong sensor output field was chosen on page 4 (Figure 7). In this particular user study, the third field of the thermometer block must have been specified as the output of the sensor. Instead, five participants chose a different field and four participants needed advice to choose the correct one. Also on page 4, a wrong metadata association (e.g., unit of measure was set to "*22.34*") happened in three cases.

On page 3 (Figure 6), a wrong connector specification (e.g., a USB connector was chosen instead of a file-based connection) and mistaken block definition (e.g., the chosen block identifier did not match the block name in the protocol) were each counted eight times. Also on page 3, mistakes in the field definition were made (e.g., not all fields of the block defined). Another mistake on page 3 was the naming of the third field in the protocol as "*22.34*" instead of "*temperature_value.*" This does not lead to an invalid SID but shows a misunderstanding of the concept. It happened four times among the nine high school students and once in Group C. Four high school students needed help from the experimenter to specify the separator signs (e.g., the experimenter had to recapitulate what separator signs are).

On page 2 (Figure 5), only one kind of mistake was made by two participants: the checkbox whether the sensor collects data was not checked.

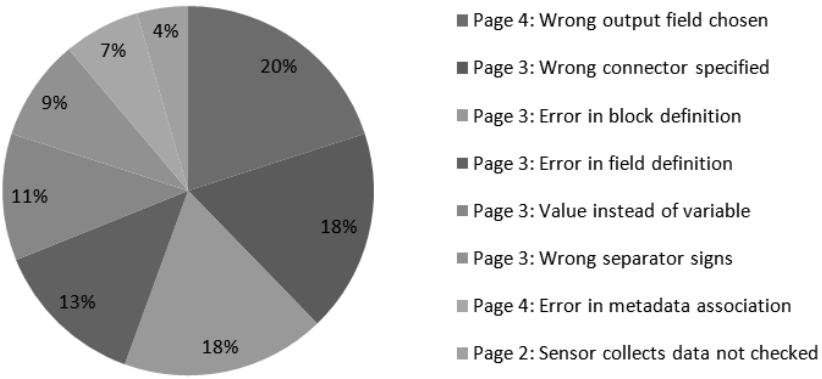


Fig. 10. Relative frequency of occurred kinds of errors

Figure 11 shows the average time per person for editing the different SID creator pages. The diagram shows that the editing of page 2 took almost the same amount of time for all groups. In Group A, B and C the editing of page 3 took the longest time which indicates that the definition of the sensor protocol structure was most difficult for them (Group A needed the most time, namely 6 minutes and 52 seconds on average). Group D put most of the time (3:47) in the definition of the sensor output and its metadata. On page 4, a common misunderstanding was that not only the actual data values but also other fields of the sensor protocol need to be defined as sensor output and uploaded to the SOS. This is not the case since the information contained in the other fields is static and specified as associated metadata by the user of the SID creator.

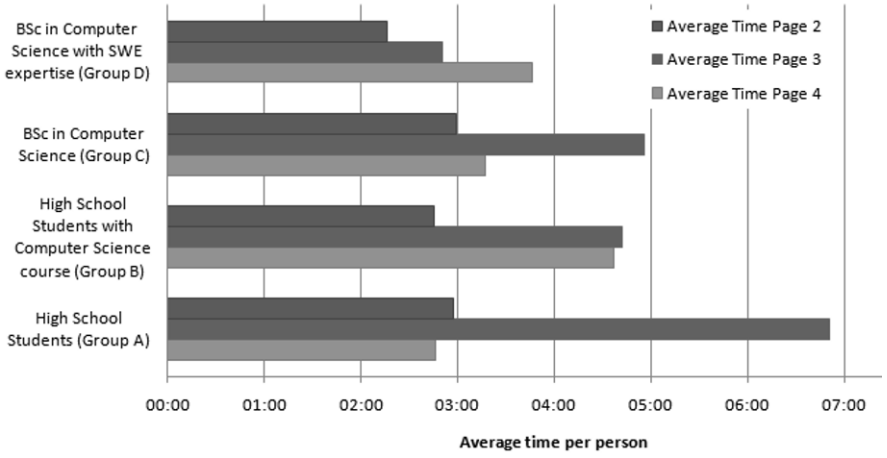


Fig. 11 Average time per person needed for editing a wizard page

After each test, the participant was asked to complete a questionnaire. For example, the participants were asked whether they think they have fully/partly/not understood the principle of the SID concept and whether it can replace manual implementation of adapters between sensor and SOS. The answer to this question should indicate a self-report measure about how sure the participant is of what he/she just did. In Group D, all five members stated to have "fully" understood the SID concept. In Group C, one person answered the question with "partly", the other five members answered with "fully". In Group B two persons answered with "fully" and three with "partly". In the most inexperienced Group A, two people stated to have "fully" understood the SID concept and the other two did "not" understand.

Additionally, the five members of Group D, who had already connected a sensor manually to an SOS, were asked whether it is easier to use the visual SID creator instead of implementing an adapter. All of them perceived the SID creator as a helpful tool for the given task and answered with yes. However, three raised the question whether the SID creator in its current design has enough functionality to support all kinds of sensor types. It was assumed that complex sensor interfaces still require a manual implementation of adapters.

7 Conclusion and Outlook

To close the interoperability gap between sensors and the Sensor Web, the SID model has been developed based on the SensorML standard. It enables the declarative description of a sensor protocol. An SID interpreter is able to translate the sensor protocol to Sensor Web protocols based on the knowledge contained in an instance of the SID model. This paper presents the SID creator which enables users to visually generate SID instances for their sensors. The SID creator prevents users from manually implementing adapters for each sensor type which shall be integrated with the Sensor Web. Instead, an additional interface description, the SID, can be created which enables the integration of the sensor with the Sensor Web. Once a sensor interface is described by an SID, it can be used in multiple applications by different user communities. Together, the SID model and SID creator are beneficial for sensor manufacturers and sensor data providers who do not have to change their sensor's protocols.

The usability and usefulness of the SID creator was evaluated by conducting a user study. The participants, ranging from high school students

to SWE experts, were tasked to create an SID for the sensors of a weather station. The analysis of the user study showed that the SID creator was very useful for the group of SWE experts. They stated that it is easier to use the SID creator to integrate a sensor with the Sensor Web than implementing an adapter manually. Four of five members of that group created a working SID. In the group of people with a BSc degree in computer sciences, who did not have experience in SWE, one third created a working SID and over 80% of those users stated to have fully understood the SID concept. Significantly higher error rates and average time consumption showed that the task was most difficult for the high school students. However, also here two out of nine people were able to create a working SID without any prior experience in integrating sensors with the Sensor Web. Finally, the results of the user study lead to the conclusion that, with increasing level of user experience, the SID creator is a helpful tool and considerably facilitates the process of sensor integration.

For the future, in particular the protocol definition page of the SID creator needs to be improved, since it caused most of the problems as the results of the user study have shown. This can be done, e.g., by enhancing the help texts and including descriptive examples, or by enhancing the current form-based user input to a more graphical design. Also, the SID creator used in this work does not yet allow configuring all details of an SID and not all sensor protocols can be defined. Further extending the SID creator to fully support the SID model will broaden the range of sensor types for which SIDs can be created.

References

1. M. Botts, G. Percivall, C. Reed, and J. Davidson, “OGC Sensor Web Enablement: Overview and High Level Architecture,” *Lecture Notes In Computer Science*, vol. 4540, pp. 175–190, 2008.
2. S. Jirka, A. Broering, and C. Stasch, “Applying OGC Sensor Web Enablement to Risk Monitoring and Disaster Management,” in *GSDI 11 World Conference, Rotterdam, Netherlands*, June 2009.
3. C. Stasch, A. C. Walkowski, and S. Jirka, “A Geosensor Network Architecture for Disaster Management based on Open Standards.” in *Digital Earth Summit on Geoinformatics 2008: Tools for Climate Change Research.*, M. Ehlers, K. Behncke, F. W. Gerstengabe, F. Hillen, L. Koppers, L. Stroink, and J. Wächter, Eds., 2008, pp. 54–59.
4. K. Walter and E. Nash, “Coupling Wireless Sensor Networks and the Sensor Observation Service - Bridging the Interoperability Gap,” in *12th AGILE International Conference on Geographic Information Science 2009*, Hannover, Germany, 2009.

5. K. Aberer, M. Hauswirth, and A. Salehi, "A Middleware for Fast and Flexible Sensor Network Deployment," in *32nd International Conference on Very Large Data Bases*, 2006.
6. A. Na and M. Priest, *OGC Implementation Specification 06-009r6: OpenGIS Sensor Observation Service (SOS)*. Open Geospatial Consortium, 2007.
7. I. Simonis, *OGC Implementation Specification 07-014r3: OpenGIS Sensor Planning Service*. Open Geospatial Consortium, 2007.
8. S. Cox, *OGC Implementation Specification 07-022r1: Observations and Measurements - Part 1 - Observation schema*. Open Geospatial Consortium, 2007.
9. M. Botts, *OGC Implementation Specification 07-000: OpenGIS Sensor Model Language (SensorML)*. Open Geospatial Consortium, 2007.
10. P. Hu, R. Robinson, and J. Indulska, "Sensor Standards: Overview and Experiences," in *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing ISSNIP'07*, Melbourne, Australia, 3-6 December 2007.
11. T. Bleier, B. Bozic, R. Bumerl-Lexa, A. Da Costa, S. Costes, I. Iosifescu, O. Martin, S. Frysinger, D. Havlik, D. Hilbring, P. Jacques, M. Klopfer, S. Kunz, P. Kutschera, M. Lidstone, S. Middleton, Z. Roberts, Z. Sabeur, J. Schabauer, S. Schlobinski, T. Shu, I. Simonis, B. Stevenot, T. Usländer, K. Watson, and K. Wittamore, *SANY - An Open Service Architecture for Sensor Networks*, M. Klopfer and I. Simonis, Eds. SANY Consortium, 2009.
12. G. Gigan and I. Atkinson, "Sensor Abstraction Layer: A Unique Software Interface to Effectively Manage Sensor Networks," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007*, 3-6 2007, pp. 479–484.
13. J. Tidwell, *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly, 2006.
14. S. Jirka and A. Broering, *OGC Discussion Paper 09-033 - SensorML Profile for Discovery*. Open Geospatial Consortium, 2009.
15. J. Nielsen, *Usability Engineering*. Morgan Kaufmann, 1993.
16. M. van Someren, Y. Barnard, and J. Sandberg, *The Think Aloud Method - A Practical Guide to Modelling Cognitive Processes*. London: Academic Press, 1994.
17. C. van Elzakker, I. Delikostidis, and P. van Oosterom, "Field-Based Usability Evaluation Methodology for Mobile Geo-Applications," *The Cartographic Journal*, vol. 45, no. 2, pp. 139–149, 2008.
18. A. Broering, S. Below, and T. Foerster, "Declarative Sensor Interface Descriptors for the Sensor Web," in *WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services*, Como, Italy, 26.-27. August 2010.
19. A. Broering and S. Below, *OGC Discussion Paper 10-134: Sensor Interface Descriptors*. Open Geospatial Consortium, 2010.

An OGC compliant Sensor Observation Service for mobile sensors

Roland Müller, Manuel Fabritius, Michael Mock

Fraunhofer IAIS, Schloss Birlinghoven, St. Augustin, Germany
{roland.mueller, manuel.fabritius, michael.mock}@iais.fraunhofer.de

Abstract. This paper presents the design and implementation of an OGC (Open Geospatial Consortium) compliant Sensor Observation Service (SOS), which supports spatial queries for mobile sensors. The work was carried out in the scope of the European ESS project, aiming at the development of a mobile geosensor network for supporting operational management in crisis events. Our sensor web service is complemented by a client programming framework which allows for browser-based access to and visualization of mobile sensors. We demonstrate several use cases, including real-time visualization of mobile sensors, spatial queries and value-based filtering. Our performance evaluation shows that the SOS implementation scales up to supporting more than 1900 sensor observations per second, which clearly outperforms other open SOS implementations.

1 Introduction

Current developments in geosensor networks show that the connection of several different sensor systems becomes important for achieving a highest possible information gain. Especially mobile sensors attract attention due to technical improvements and falling prices, including mobile phones with integrated sensors or flying devices like unmanned aerial vehicles (UAVs) or quadrocopters. Our work has been performed in the context of the EU-funded research project "ESS - Emergency Support System" with the scenario being a disaster recovery application, in which many mobile sensors are connected to a geosensor network.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_9,

The ESS project was started in June 2009 with the goal of setting up a portable crisis management system. Stationary or mobile sensor platforms like UAVs collect environmental sensor data, which is sent in real-time to a central server. The fusion of the collected data may then assist the emergency case operator in deciding on how to setup a rescue plan (Algosystems 2009). Due to the instantaneous access to the sensor's measurements, the operator can immediately detect changes in the environment and refine his decisions. As ESS aims at providing an extensible, open Platform, use of the OGC sensor web standards are envisaged.

Most of the mobile sensor tracking services currently exist as proprietary solutions for specific sensors. Taking mobile phones as an example, there are system-dependent applications like the *Nokia Sports Tracker* (Symbian) or *Google Latitude* (Android). To combine or exchange the collected sensor data, we need to employ standards that allow the extensibility towards more, arbitrary sensors and enable more re-usable software development for large scale sensor web communities. The *Open Geospatial Consortium* (OGC) has established such standards for geoservices from which especially the *Sensor Observation Service* (SOS) (OGC SOS 2007) is interesting, being a web service that enables the exchange of sensor metadata and instantaneous access to the sensor's observations.

The challenge addressed in our paper is to make the SOS applicable for large scale mobile sensor data as the SOS standard and current implementations primarily concentrate on supporting stationary sensors. Furthermore, we want to provide ready-to-use components, which support the design of geospatial web services. This includes servers for data storage as well as end-user front-ends that are designed according to existing web standards, making them accessible via web clients running in standard browsers. A main focus lies on improving the SOS regarding the support of mobility and to allow the client to perform spatial queries with respect to actual sensor positions. Another aspect is to handle the "XML overhead" from the programming and performance points of view.

Our approach is to define a subset of OGC functionality for supporting an extensible set of mobile sensors. We add mobile location information in the appropriate response documents of the SOS, and extend its functionality by an option to retrieve the latest observation from a specific sensor. The compact design of the underlying database structures was optimized for the aforementioned functionality and a ready-to-use web client framework developed for wrapping the access to the SOS.

Section 4 shows performance tests, in which we demonstrate the scalability of our implementations. We also compare the SOS with a similar open source development as well as with our own raw data transmission protocol.

2 Related Work

The SOS specification is split into three profiles, from which the *core profile* is the only mandatory one. It contains three operations, being *GetCapabilities* for requesting OWS metadata (OGC WSS 2007) of the SOS web service, *DescribeSensor* for acquiring the sensor description of a specific sensor, mostly in *SensorML format* (OGC SML 2007), and *GetObservation* for requesting actual measurements in *Observations & Measurements* (*O&M*) format (OGC OM 2007). Operations for writing new data into the SOS are contained in the optional *transactional profile*. Its *RegisterSensor* request sends a *SensorML* document to the SOS for publishing a new sensor and the *InsertObservation* operation allows sending new measurements in *O&M* format.

So far, existing SOS implementations are designed as special purpose systems which cannot be used as building blocks for other systems or they only implement the SOS *core profile* and lack the possibility of inserting new observations by not supporting the *transactional profile,* e.g. deegree (OSGeo 2010-1), MapServer (OSGeo 2010-2), OOSTethys (OOSTethys 2010). An overview of their supported operations is shown in Table 1.

Table 1. Comparison of SOS implementations

Name	Implements
Degree	Core profile
MapServer	Core profile, Describe ObservationType
OOSTethys Java SOS Toolkit	Core profile
52°North SOS	Core & Transactional profiles, GetFeatureOfInterest, GetResult

Other open SOS implementations suffer from performance problems or do not support mobility in a standard-compliant manner (e.g. 52°North). For the 52°North implementation, there is a mobility extension of the standard which is described in (Stasch 2008). Here, the observation model is extended by two elements, namely a *SamplingFeature* and a *DomainFeature*. The former represents the position where the measurement took place; the latter describes the area in which the sensor is moving (e.g. a lake). This is realized by making modifications to the existing SOS operations and by inventing a new *UpdateSensor* operation, which has to be invoked whenever the sensor position changes. So each new measurement from a moving sensor requires the invocation of two requests, namely *InsertObservation* for the new measurement values and *UpdateSensor* for the new position. If a client of the mobility-extended 52°North SOS wants to

track a mobile sensor's position over a specific period, it has to invoke lots of *DescribeSensor* requests. For all points in time during that period, the according position must be requested individually.

Regarding the user front-end, we aim at a similar goal as the uDig SDK (Refractions Research 2010),. which provides components for the development of specific client applications, including a plug-in for SOS support (Priess and Kiesow 2010). In contrast to uDig, our client can be executed in web browsers.

On the other hand, there are also projects that use a server-side integration of web-map services to pre-process, i.e. the OGC sensor information, in order not to overload the client. The *OOSTethys* project (SURA 2010) developed a server that supports the OGC SOS standard. They are also using a lightweight GoogleMaps client to visualize their stored information in a very basic way. The diploma work of Riegger (Riegger 2006) describes a way to access sensor information in a web-based way, but they are not using the OGC SOS standard as we and OOSTethys do.

Alternatively, *OpenLayers* (OpenLayers 2010) provides a framework for client-based integrations of different data sources and layers. It is a web-mapping library that is used by the OpenStreetMap project to access OGC standards for their web-mapping purpose. The anonymous community developer *Bartvde* developed a lightweight SOS interface with JavaScript. It supports a rudimentary SOS integration. A similar approach by Dominik Helle (Helle 2010) also uses OpenLayers. His *SensorGIS* is able to handle basic requests and can show the received information in tables and charts.

The aforementioned projects only support rudimentary tools for visualization (e.g. sensor mapping) or no web-browser integration. Rich visualization for generic sensors with spatial references is not yet available for browser-based clients because these implementations are too sophisticated for web-applications. Using the OGC standard, the clients would be able to access different OGC sensor servers without further adjustments.

3 Architecture

Our application consists of two main parts, namely the Sensor Observation Service and the web client framework, which are both presented below.

To access and visualize the information of the SOS, a JavaScript-based *Client Framework* has been developed. The framework can access an interface implementation of the *OGC SOS* standard. Ready-to-use widgets are capable to visualize data series in charts and show spatial information

on a map. The whole framework and SOS interface was implemented with JavaScript.

In Figure 1 a rough overview of the system is shown. The whole system is developed in JavaScript and uses OpenLayers as web-mapping library. The framework adds support for the OGC SOS standard and visualization features.

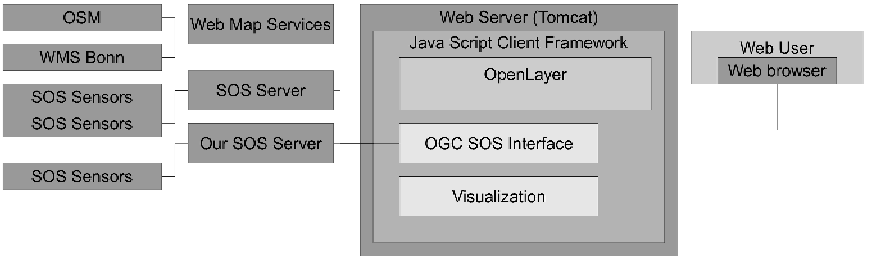


Fig. 1. Overview on the rich client framework interacting with OGC services

3.1 SOS - Design

To give a first overview of the SOS and the data generating sensor clients, the structure is depicted in Figure 2.

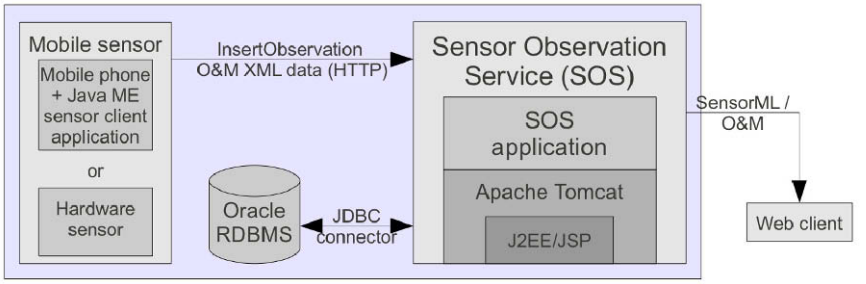


Fig. 2. SOS application structure

On the left side, the data source which creates new measurements is presente d in the form of a mobile phone or another mobile sensor device, determining its current position via GPS. On the phone, a Java application links the position data with further measurements, which it may get from an acceleration sensor. This observation is then sent in an OGC-compliant way to the SOS via the *InsertObservation* request.

The SOS is implemented as a servlet and runs on an Apache Tomcat servlet container. It connects to an Oracle RDBMS for storing and retrieving the sensor metadata and measurement values.

An external web client can connect to the SOS and use OGC-compliant XML-requests to fetch server and sensor metadata or measurements, as well as insert new sensors or observations into the web service.

For being able to compare the SOS performance with a raw data transmission, a stand-alone TCP/UDP Java server application was written. It stores the received raw data values in the same database as the SOS server does.

3.1.1 Supporting mobile sensors

The OGC standards were primarily defined for stationary (also called "in-situ") sensors. Here we describe how we accommodate mobile sensors into the OGC standards without violating their compliance.

A major problem for coping with mobile sensors is the lack of a function that allows updating the position information. The *DescribeSensor* SensorML response normally returns the sensor location which the server got once during the initial sensor registration in the *RegisterSensor* request. Instead of implementing an additional *UpdateSensor* operation as proposed in Stasch (2008), which is not part of the standard so far, our approach is to transmit the mobile sensor's current position inside the "feature of interest" of an *InsertObservation* request.

In the *GetObservation* O&M response, the location parameters (latitude, longitude, altitude) are treated as phenomena and listed together with the other sensor measurements. By replacing the registered position with the most recent one during a *DescribeSensor* request, the response stays fully compliant in terms of XSD schema validation.

The advantage of our approach is that due to declaring the position parameters as phenomena, we enable the client to request the whole track with only one *GetObservation* request. This also enables an easy usage of temporal and spatial filters.

3.1.2 Feature of interest

The "feature of interest" (FOI) is the object for which an observation is made. According to the OGC SOS specification, the sensor must indicate it for each observation, but the semantic interpretation is left open to the SOS implementations. In many cases, mobile sensors do not necessarily know whether they are currently above a lake, a forest, etc., making it difficult to determine the FOI. Also, the detection of the current street name would re-

quire further intelligence by employing a map-matching algorithm. Regarding mobile sensors, there arises the problem that if the mobile sensor passes a lot of different FOIs during its tour, this would heavily bloat up the *GetCapabilities* document, as they are all listed there for each sensor. Thus the decision was made to set the "feature of interest" equal to the *procedure*, that is the sensor ID. If there should arise a need for a different FOI later on, this is implementable by adding an additional column to the sensors or measurements database table.

3.1.3 Observation offering

A further ambiguous situation arises with the "observation offering", which combines several related measurements into a group and which is the first required parameter in the *GetObservation* request. There are different criteria for the grouping and they are mostly dependent on the intended use of the measurements. The 52°North server for example defines offerings equal to the phenomena. This facilitates the request of attributes like temperature or wind speed over a large area that contains many sensors. However, if a client wants to receive all measurements of a specific sensor/procedure, it would have to send multiple requests, one for each phenomenon. A different choice is to set the offerings equal to the procedures. This allows an easy retrieval of all phenomena for a specific sensor, but on the other hand it requires multiple requests if the client needs the temperatures of a large area containing several sensors.

As the sensor template document from the ESS project suggests choosing the latter variant, our implementation will also use the *procedure* as the offering.

3.1.4 Further restrictions

Since we only want to cope with physical sensors, we assume that the sensor inputs are always equal to the outputs, i.e. there is no visible preprocessing on the sensor itself. This reduces the database structure in which the sensor metadata is stored.

Officially, the *timePosition* parameter in the *GetObservation* request only allows for an ISO 8601 timestamp for requesting observations of a certain point in time. Inspired by the 52°North implementation, a special *latest* keyword allows requesting the newest measurement of a specific sensor. This is useful for obtaining the last known sensor position, including the related values for all its phenomena.

Despite the standardization, there are two different methods for indicating the current sensor location. In the *InsertObservation* example of the

OGC SOS standard, the position is stored in the result section, thus being a phenomenon. The 52°North examples as well as the template of the ESS project put the position into the "feature of interest" section. As we want to stick with the template, we also chose the second alternative.

3.1.5 SOS database design

The database will run on a different machine than the servlet container. This should result in a higher performance of the whole system, as each machine can use the full power for its dedicated function. Also due to the costly license of the Oracle RDBMS, one dedicated database machine where multiple application servers can connect is preferable.

For facilitating the implementation of the spatial filters, which are needed to query sensors that are located within a specific area, the positions must be stored as spatial geometries. Further, this enables us to visualize the sensor locations with the uDig GIS for validation purposes.

3.1.6 Batching

A performance improvement is available only for queries that write data into the database. A stored procedure was written, which is a function that is executed on the database side and contains the main query with all required sub-queries in our case. For the JDBC client, it appears as one database function. Instead of waiting for the number of batched queries to be reached, it is possible to explicitly induce them to be sent. This is done with the *sendBatch()* method of the *OraclePreparedStatement*. In our case it is executed in the function that closes the database connection on server shutdown.

3.1.7 Database schema

The database schema was created with the *Oracle SQL Developer Data Modeler* under consideration of the database normalization criteria to avoid redundancies. An overview of the tables in form of an ER-diagram is shown in Figure 3.

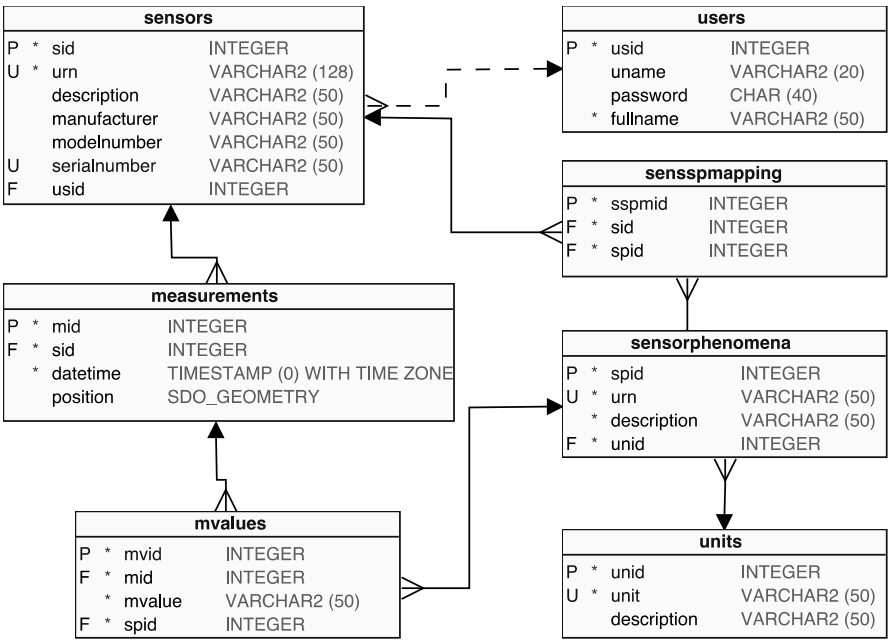


Fig. 3. Entity relationship diagram of the database layout

In the *sensors* table, the mandatory sensor ID and its optional metadata are saved. A foreign key from the *users* table can specify which user is responsible for the sensor. In *sensorphenomena*, all phenomena are stored, with each of them referencing to their appropriate unit of measurement from the *units* table. The purpose of the *senssmapping* table is to map an individual number of phenomena to each sensor.

For storing the observations, the *measurements* table was created. It must contain the sensor ID, which it was received from, and the corresponding timestamp. The position is contained in a spatial geometry for which a spatial index was created. As the number of measurement results may vary for each sensor and observation, they are stored in the additional table *mvalues*, which also have a mapping to their appropriate *sensorphenomena* entries. The measurement values are stored as strings without distinguishing between data types like integer, float, etc. This facilitates the implementation, and as the input and output of the SOS are always XML text documents, a differentiation is unnecessary.

During the implementation of the spatial filter queries it was observed that in spite of using a spatial index, the response time was rather long. When directly executing the spatial queries on the database, we sometimes received Java exceptions if invalid parameters were used. This results from

the fact that the *Oracle Spatial* extension - in opposite to the database core - is written in Java, which may be the cause of the response delay. So the spatial operator that is needed to select sensors within a certain area is only used for areas, which are actually defined as polygons. If, as a special case, a rectangular bounding box is specified as a spatial filter, we perform a manual range comparison of the specified coordinates.

3.2 Client framework - design

Figure 4 gives a short overview on the components of the client framework.

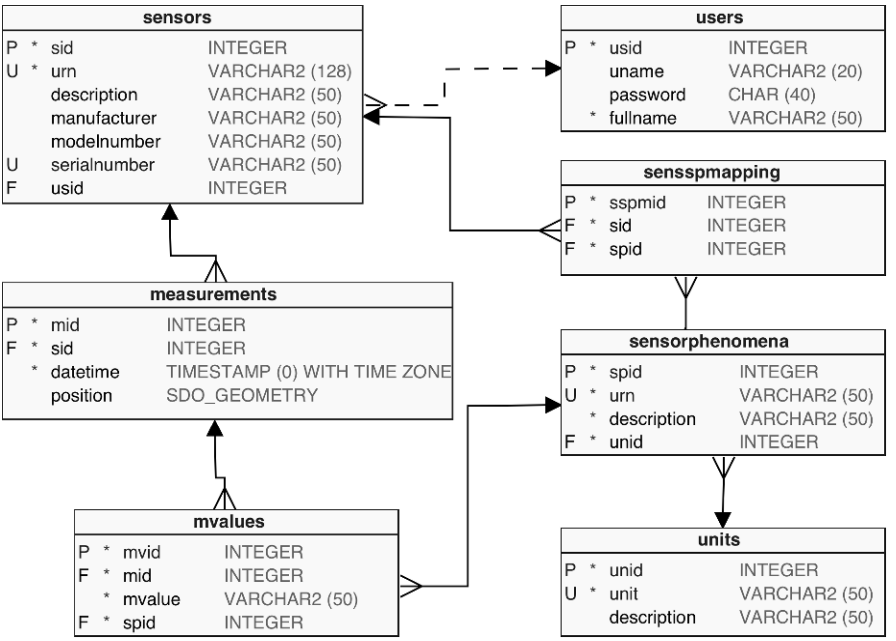


Fig. 4. Framework architecture

To simplify the distribution, all components have been implemented in JavaScript. Besides OpenLayers as a web-mapping library, JQuery was used for comfortable DOM manipulations. The client framework will be used to build rich Internet client applications and provides a variety of *widgets*. These *widgets* are not hard-wired to each other; it is possible to use only a subset of the available *widgets*. For this purpose, an *observer pattern* was used to publish any changes. *Widgets* can register to an *event*

provider which publishes the information to all registered *widgets*. Using such an *event-system,* programmers of new widgets only have to know what events are used by the existing widgets.

Data, which can be provided by any external SOS, is handled by the framework in order to keep a single consistent copy of the data. Additionally, the data is restructured in a *data model* that every widget understands. An extra *DataAdapter* handles that task. This also makes it possible to integrate more interfaces (other than SOS) into the framework. In this case, only a new *DataAdapter* for the specific data source must be added. Updates of existing data are handled by a *DataUpdater,* which automatically polls the Sensor Observation Services in the background for new data. If updates from another data source are integrated, an updater must be added.

3.2.1 Visualization widgets

Basically, there are two ready-to-use widgets for visualization. Both are working very closely together to maximize the interactive behaviour. They are designed to show data with spatial references and visualize the data such that the important information can be utilized in an efficient way. Data without a spatial reference can also be shown, but the interactive features are limited. On the one hand, the data series can be shown in charts. For this purpose, the *JQuery* plugin *Flot* (Laursen 2007) was used. Single data series can be en- or disabled and sensors' positions can be marked on the map. If the sensors are mobile, a track can be built and shown. In addition, range filters can be applied to the chart to show only partial tracks, where the measurements are within the selected value range (see Figure 7 in subsection 4.1.4)

To also support data sources where each data set represents a single feature with specific coordinates, the cluster-strategy (OpenLayers 2010a) of OpenLayers is used. The advantage of this technique is the clustering of many spatially close features into a single item. This can show thousands of interactive features on the map.

In addition, *range filters* can be applied to the chart, which can be shifted and resized. The according features on the map will directly be updated. By moving the filter on the time-axis, an operator has the opportunity to replay the behaviour of phenomena. By specifying a filter on the y-axis, critical information can be selected and highlighted, for example the toxic concentration of the target area. In Section 4.1, different use cases are shown.

4 Evaluation

4.1 Use Cases

4.1.1 Mobile sensors

One use case works on the basis of a chemical sensor developed in the ESS project. It provides facilities to estimate the expansion of a toxic cloud by measuring attributes such as wind direction, wind speed, and gas concentration. Several of these sensors were placed at a fixed position or mounted on moving vehicles at the April 2010 field trial of the ESS project to indicate toxic incubation of the surrounding area (using an alcohol-soaked cloth for generating reasonable sensor values).

The sensors have completely been modeled in SensorML and O&M as a system providing five different phenomena. The whole SensorML specification takes more than six pages and is out of the scope of this paper, but will be available to the public in the corresponding report of the ESS project.

For being able to perform an outdoor test with a mobile phone sending data to the SOS, we set up a Tomcat server, which is accessible through the Internet. The mobile phone then sends its position and further measurement values (e.g. network strength, acceleration sensor) via UMTS or GPRS to the SOS servlet.

4.1.2 Real-time visualization

Sensors are periodically uploading new measurements to the SOS. Mobile sensors additionally change their position. The SOS standard does not support an update function for the client to be informed automatically when new measurements are made.

To update the user-interface with new data, the client-framework requests to the server in a background thread. A user-defined update interval can be specified. The requests contain a special filter, which allows requesting the latest observation only. The updater of the client framework directly updates the data model that already contains previously received data. In addition, all widgets are informed about the new data by the event system.

Note that this ready-to-use approach risks missing one or more measurements or receiving the same measurement multiple times. Alternatively,

a more complex filter can be used that requests all observations made after a specific time. In this case, management of the requested time must be implemented explicitly by the programmer.

4.1.3 Large scale analysis (Flickr)

In a companion project (Kisilevich et. al. 2010), the locations of geo-tagged Flickr photos have been visualized with the appropriate data already available in one big CSV file. For being able to access this data using our SOS, an import tool was written. We created *one* single sensor for which each inserted measurement represents a geo-located Flickr photo, without distinguishing between the different Flickr users who produced the images. Besides the location and timestamp of each picture, its URL and the description were defined as phenomena.

Figure 5 shows these data using the clustering strategy which was already described in Section 3.2.1. This method visualizes how many people have made pictures (and uploaded them) at specific places. The hotspots are shown on the map - in this case they are from Berlin, using a time-frame of two months.

4.1.4 Spatial, temporal, and result filters

As already mentioned before, the SOS supports spatial and temporal filtering. Both can be used in the client framework, which additionally allows the application of range filters to the measurement values.

The application of a spatial filter is shown in Figure 6. If the user draws a polygon on the map, the coordinates of its corners will be sent in a *GetObservation* request to the SOS which only returns the observations that are located within the specified area.

In the previous Flickr example (Figure 5), we displayed measurements of a two months' range by using the appropriate temporal filter in the SOS request. The returned results were further filtered in Figure 6 by selecting a range of several days, as it is shown in the plot window.

Figure 7 shows a chart within which a *range filter* is applied for selecting the sensor values to be displayed. Based on this filter, the coordinates where the measurements are within the specified range are shown in a different color on the map. With this technique, the interesting part of a track can directly be visualized.

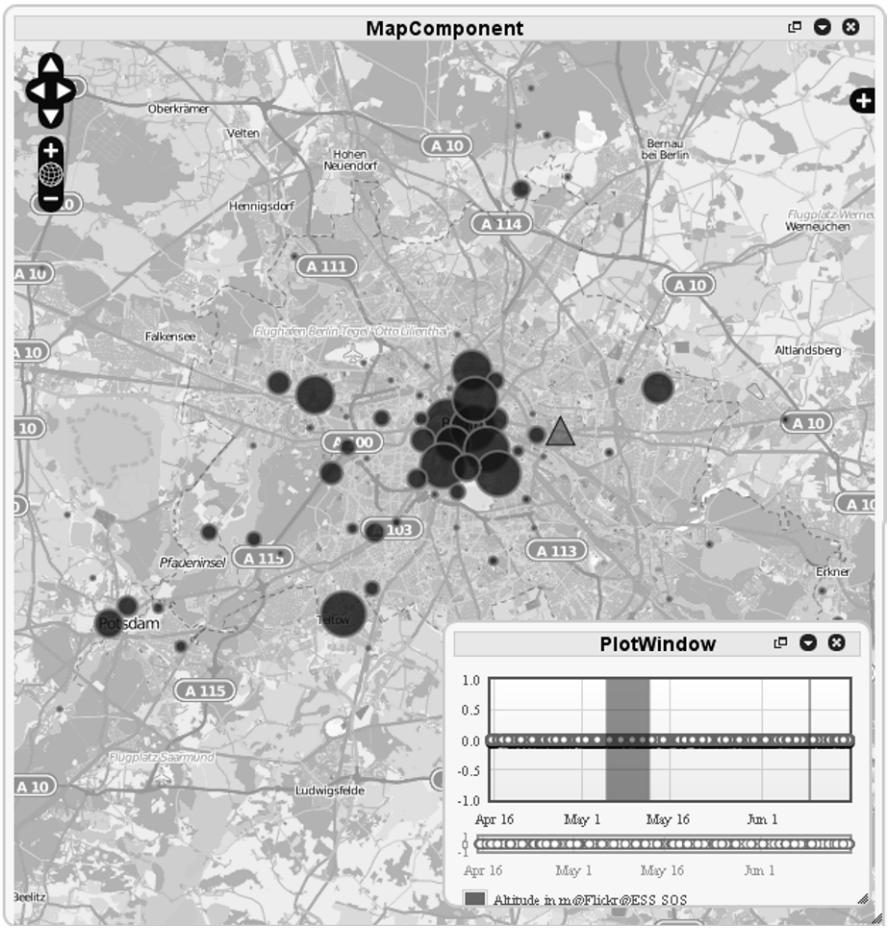


Fig. 5. Using the time filter, a more specific time interval can be shown

4.2 Performance

For HTTP performance tests, there are tools like the Apache *JMeter* which is written in Java and the *Apache HTTP server benchmarking tool* (abbreviated *ab*) as a C application. According to Brittain (2007, p, 129), *ab* is able to perform more requests per second than *JMeter* for which reason the former should be preferred. The same source also advises to start the load testing tool on a different machine than the web server is running on.

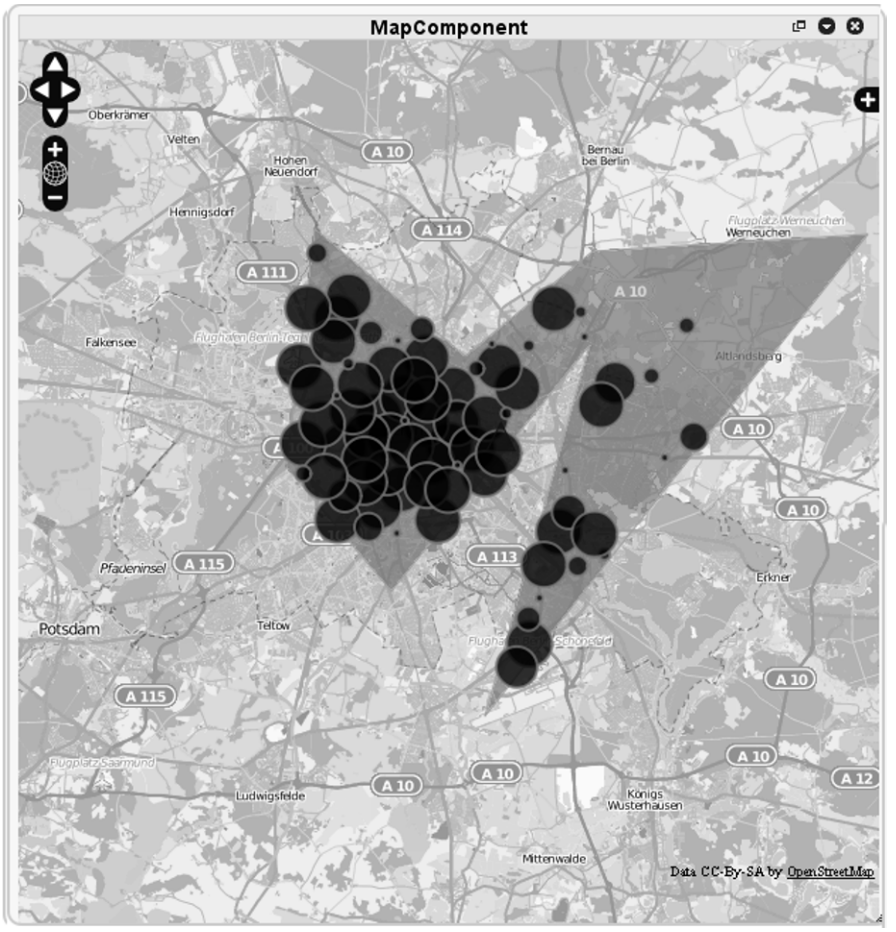


Fig. 6. A spatial polygon filter applied to the Flickr information

4.2.1 Apache Benchmark

The *ab* tool was used in version 2.0.40-dev for performing *InsertObservation* requests on our SOS implementation, with each observation containing two phenomena. It was configured to do 100 concurrent requests and 10000 requests in total (the corresponding command line parameters are *-c 100* and *-n 10000*) The server was started in varying configurations by toggling the schema validation and the batching, which was set to 500 queries. Each test configuration was benchmarked 5 times, from which we picked the best result (variance was low). An overview of the configurations and the test results is shown in Table 2.

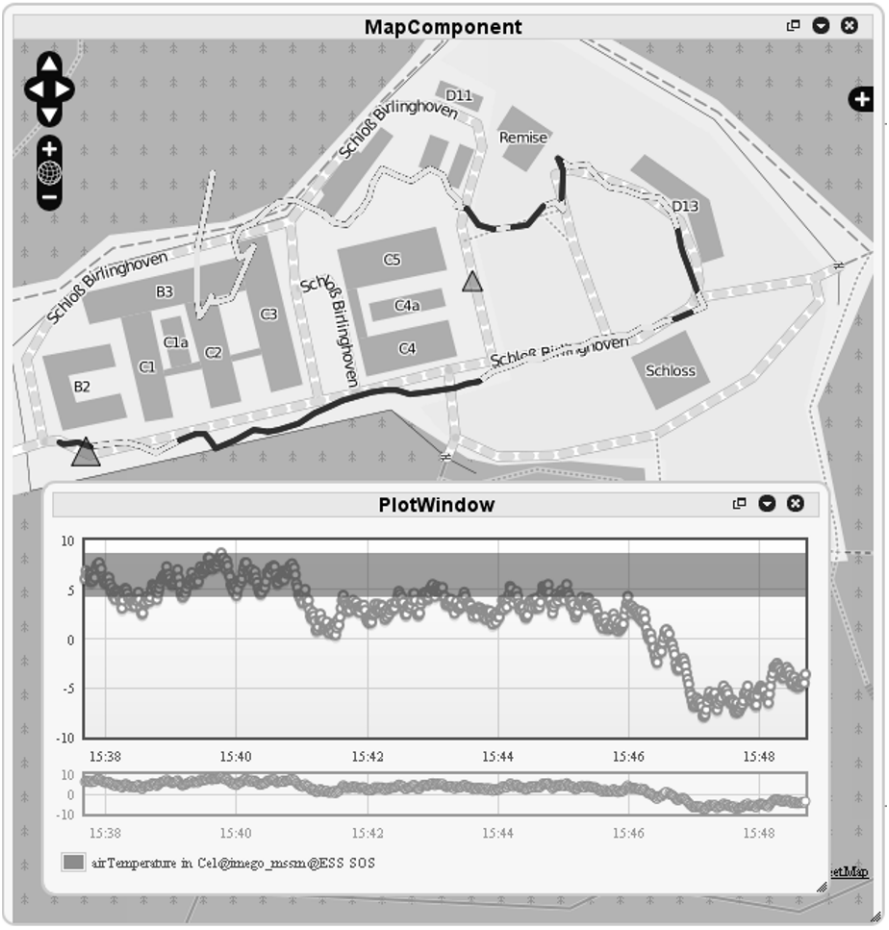


Fig. 7. The coordinates where the temperature is higher than 5 are highlighted

Table 2. Apache HTTP server benchmarking tool black box test

Validation	Batching	Throughput (Requests /s)
On	Off	32,86
Off	Off	150,47
On	On	23,19
Off	On	2041,67

From the results we can see that the schema validation has a huge impact on the performance: with batching disabled, turning off the validation improved the throughput by a factor of 4.5, with batching enabled we even got an improvement by factor 88.

Looking at the batching option on activated schema validation, we get a small slowdown when batching is enabled. This low throughput rate t is presumably due to the higher administrative effort that the JDBC driver needs to cache the queries. But when the schema validation is disabled, the activation of batching causes a speedup of factor 13.5.

Regrettably we were not able to use the Apache benchmark tool on the 52°North implementation as the server only accepts one observation per second for each sensor, and the *ab* tool cannot modify the content of the input XML file, allowing us to specify dynamic sensor IDs and timestamps.

4.2.2 Synthetic load generator test

We implemented a multi-threaded Java test client, allowing us to compare the performance of the different servers (our SOS, raw data transmission and 52°North SOS) under equal conditions. It simulates a large number of sensors by creating a thread for each client. We started it multiple times with the number of threads increasing in steps of 50 until reaching 2000 concurrent threads. Every thread did 10 *InsertObservation* requests one after another, from which each of them was allowed to have a maximum duration of one second, as this is the common interval in which GPS devices update the position. In addition to this duration violation, we also counted packet loss, meaning that the client got a connection exception and the packet did not reach the server at all. The tests were performed for all four SOS configuration combinations (enabling/disabling batching and schema validation). The average duration of a request for each of the transmission options is depicted in Figure 8.

Looking at the different SOS configurations, the test confirms the results of the previous Apache benchmarking tool. With schema validation activated and the batching option not significantly causing a difference, the server completely fails at doing more than 150 threads in parallel. In the Tomcat log file there were out of memory errors, as the server cache fills up with more incoming requests than it is able to process.

With schema validation and batching both turned off, the first timing constraint failures also occur at 150 threads, but there were no memory errors and the server still kept up at 700 threads after which the first packet losses happened.

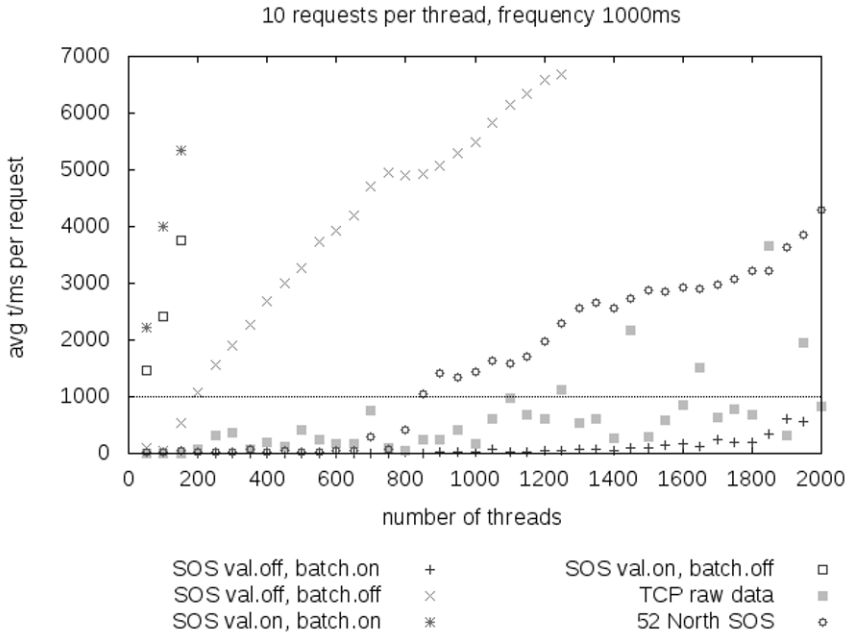


Fig. 8. Java test client throughput results

The most performing SOS configuration is the one with schema validation turned off and batching enabled. The first duration violation occurs at about 1000 concurrent threads and slowly starts to rise until 1900 threads, where we have an increasing slope and also the first packet loss. This number is also not far away from the maximum of 2000 requests per second from the Apache benchmark.

A comparison with the 52°North SOS shows that its first duration violation occurs at about 800 threads in parallel. This lies halfway in between the two options of batching enabled and disabled of our implementation with both having schema validation deactivated.

Switching over to the raw data transmission, the first TCP duration violation already occurs very early, but further duration violations raise with a low slope and there is only marginal packet loss up to 2000 threads.

To answer the initial question on the overhead that the SOS *InsertObservation* XML request via HTTP causes over a raw data transmission, we can conclude that depending on the server configuration, it does not make a big difference. In the TCP case, the response time was even higher than with the fastest configuration of our SOS implementation. This is presumably due to the high optimization level of the Tomcat implementation

and its multi-threading architecture that our raw data Java server is missing.

4.2.3 Client Benchmark

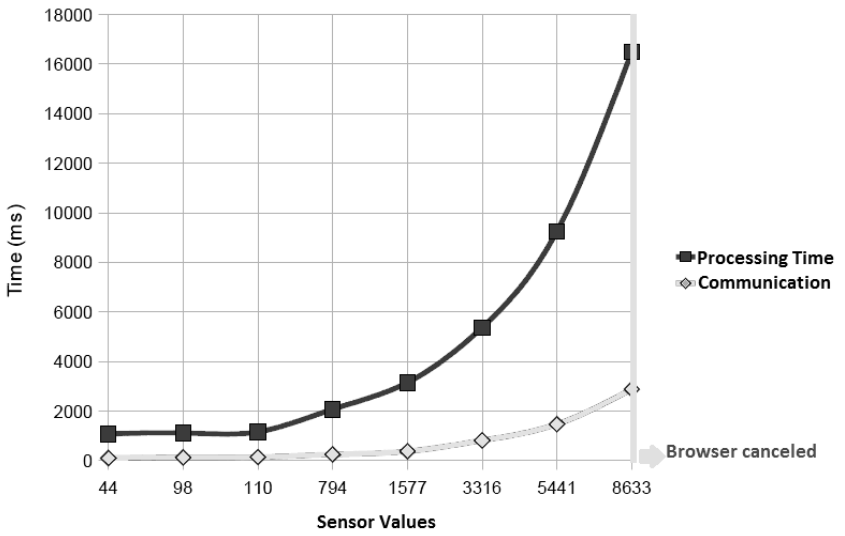


Fig. 9. A benchmark of client framework's processing and response-time

Figure 9 shows the results of a benchmark for the client-framework. The benchmark was done with Firefox 3.6 on an Intel Core2Duo E8400 (3 GHz). The grey line shows the time the SOS needs for the response. The black line shows the processing time of the received data. The processing includes the XML parsing, interpreting, and plotting into the chart. The computational complexity seems to be exponential, which might be due to JavaScript's interpretive way of accessing variables. In addition, all browsers interrupt JavaScript processing in the case of a computation taking too long. The Firefox standard settings interrupt a script after working five seconds on the same function. In our case, Firefox stopped after ~16 seconds working on a request that contains 50600 single samples. During this time, it ran through different parts of the program and suddenly stopped. On the aforementioned machine, the limit for Firefox is about ~8000 requested observations. Each observation consists of 6 measurements (including timestamp) Thus about 48000 samples can be requested with this framework. The chart widget seems to be most time consuming. It needs the date in a format of time/value pairs. Thus, the whole received data

must be rearranged and is being extended in memory size by approximately 50%.

To improve the performance, we tested an experimental client-framework version using JavaScript *webworkers* (JavaScript threads). This version only worked on the newest *Firefox alpha* (Minefield), and encountered other limitations (probably due to the alpha status)

In general, there will always be a point where the received data is too much to be handled by a browser. Perhaps, JavaScript and browsers are, at the time of writing, a limitation for *Rich Clients*. In a real emergency situation, it is not acceptable that a script suddenly stops working. Browsers, when dealing with large data sets in JavaScript, lack robustness and are not scalable. The data must be limited somehow, or data sets must be pre-processed by another server component.

5 Conclusions

Our implementation showed that an OGC-compliant interpretation of the SOS standard is possible in such a way that mobile sensors can be meaningfully supported. By adding location information to every observation as a phenomenon, a geo-spatial interpretation and analysis of sensor data becomes possible. In the simplest case, tracks and actual positions of sensors can be visualized in a portal in an OGC compliant manner. Furthermore, spatial queries for historical sensor data are supported by this usage of the SOS standard and by our SOS implementation.

Even in our single-server implementation, a throughput of more than 1900 *InsertObservation* requests per second were achieved, which is sufficient for the scope of the ESS project. XML encoding of sensor data turned out not to be a significant limiting factor in the performance compared to a proprietary raw-data submission scheme. Hence, we can profit from the extensibility and compatibility offered by relevant OGC standards for data formats (SensorML, O&M) without sacrificing performance.

The JavaScript-based client framework we developed supplies ready-to-use chart and map widgets for fulfilling the requirements of basic GIS functionalities. It integrates OpenLayers and provides functions for sensor mapping and track visualization. Different techniques have been developed to illustrate the received data. Using the spatial, temporal, and range filters on measurement data recorded by mobile and/or stationary sensors, a fast analysis of actual situations is possible. By using clustering, also large amounts of data can be visualized in a clear representation.

Improvements can be made on the SOS side in terms of performance by employing caching. To reduce latency for better supporting real-time updates, the "latest" sensor values could be stored in SOS RAM, thus avoiding internal database queries for this special case. For both the SOS and the client framework, a limitation of data to be transmitted in a single response would be useful (similar to the SetFetchSize primitive available in some SQL implementations). Currently, the server and the web browser can become unresponsive if too many measurements have to be returned by the SOS. We will evaluate the upcoming SOS 2.0 standard in this respect. The client framework's performance can be improved by using new technologies like *webworkers* and *Direct2d* which unfortunately are currently not supported by all operating systems and web browsers.

Acknowledgements

This work has been partially supported by the EU under grant No. 217951 (ESS project, "Emergency Support System").

References

Algosystems S.A. (2009) Press release at the Start of ESS. http://www.ess-project.eu/ downloads/category/1-press.html, August 2009.

Jason Brittain and Ian F. Darwin. Tomcat: The Definitive Guide. O'Reilly Media, 2007.

Dominik Helle (2010) SensorGIS. http://maps.terrestris.de/sensorgis/.

Slava Kisilevich, Milos Krstajic, Daniel Keim, Natalia Andrienko, Gennady Andrienko (2010) Event-based analysis of people's activities and behavior using Flickr and Panoramio geotagged photo collections, IV 2010. 14th International Conference on Information Visualization, London, UK, July 2010.

Ole Laursen (2007) Flot – Attractive Javascript plotting for jQuery. http://code.google.com/p/flot/, 2007.

OGC OM (2007) Open Geospatial Consortium Inc. Observations and Measurements -Part 1 -Observation schema. OGC 07-022r1, December 2007.

OGC WSS (2007) Open Geospatial Consortium Inc. OGC Web Services Common Specification. OGC 06-121r3, February 2007.

OGC SML (2007) Open Geospatial Consortium Inc. OpenGIS Sensor Model Language (SensorML) Implementation Specification. OGC 07-000, July 2007.

OGC SOS (2007) Open Geospatial Consortium Inc. Sensor Observation Service. OGC 06-009r6, October 2007.

Open Layers (2010) Open Source Geospatial Foundation. OpenLayers. http://openlayers.org/.

OpenLayers (2010a) Cluster strategy example http://openlayers.org/dev/examples/strategy-cluster.html 2010.

Carsten Priess and Martin Kiesow (2010) uDig SOS Plugin. http://52north.org/SensorWeb/clients/uDig_SOS_Plugin/index.html.

Uwe Riegger (2006) Interdisziplinäre Referenzimplementierung eines drahtlosen Sensornetzwerkes zur Erfassung von Messdaten und deren simultanen Speicherung, Auswertung und Visualisierung in einem WebGIS. http://www.terrestris.de/wp-media/downloads/ Diplomarbeit_Riegger_2006.pdf, Hochschule Karlsruhe 2006.

Christoph Stasch, Arne Bröring, and Alexander C. Walkowski (2008) Providing Mobile Sensor Data in a Standardized Way – The SOSmobile Web Service Interface. Short Paper on 11th AGILE Conference. Girona, Spain, 5-8. May 2008.

SURA (2010) SURA and Marine Metadata Interoperability 2010. OOSTethys. http://www.oostethys.org/.

OSGeo (2010-1) deegree. http://www.deegree.org/.

OSGeo (2010-2) MapServer. http://mapserver.org/.

OOSTethys (2010) JAVA SOS Toolkit. http://www.oostethys.org/downloads/oostethys-toolkit-java.

Refractions Research (2010) uDig. http://udig.refractions.net/.

Empirical Study of Sensor Observation Services Server Instances

Alain Tamayo, Pablo Viciano, Carlos Granell, Joaquín Huerta

Institute of New Imaging Technologies
Universitat Jaume I, Castellón de la Plana, Spain
{atamayo, pablo.viciano, carlos.granell, huerta}@uji.es

Abstract. The number of Sensor Observation Service (SOS) instances available online has been increasing in the last few years. The SOS specification standardises interfaces and data formats for exchanging sensor-related information between information providers and consumers. SOS, in conjunction with other specifications in the Sensor Web Enablement initiative, attempts to realise the *Sensor Web* vision, a worldwide system where sensor networks of any kind are interconnected. In this paper, we present an empirical study of actual instances of servers implementing SOS. The study focuses mostly in which parts of the specification are more frequently included in real implementations and how exchanged messages follow the structure defined by XML Schema files. Our findings can be of practical use when implementing servers and clients based on the SOS specification, as they can be optimized for common scenarios.

1 Introduction

The Sensor Web Enablement (SWE) initiative is a framework that specifies interfaces and metadata encodings to enable real time integration of heterogeneous sensor networks into the information infrastructure. It provides services and encodings to enable the creation of web-accessible sensor assets (Botts et al. 2008). It is an attempt to define the foundations for

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_10,

the Sensor Web vision, a worldwide system where sensor networks of any kind can be connected (van Zyl et al. 2009).

SWE includes specifications for service interfaces, such as:Sensor Observations Service (SOS), a standard interface for requesting, filtering, and retrieving observations and sensor system information (OGC 2007); and Sensor Planning Service (SPS), a standard for requesting information about the capabilities of a sensor and for defining tasks over those sensors (OGC 2007a). It also includes encodings for the information exchanged between information providers and consumers. The main encodings are Observation and Measurement (O&M) (OGC 2007b), which defines standard models for encoding observations and measurements from a sensor; and the Sensor Model Language (SensorML) (OGC 2007c) defining standard models for describing sensor systems and processes. The format of the exchanged messages is defined using XML Schema, a language used to assess the validity of well-formed element and attribute information items contained in XML instance files (W3C 2004, 2004a).

The number of SOS server instances available online has been increasing in the last few years. Although these instances are based in the same implementation specification, they frequently differ in subtle ways of representing information, for example, which subsets of the schemas they use, which protocols are used to request information, etc. These differences make *interoperability* a goal that is hard to achieve in practice.

In this paper, we present an empirical study of servers implementing SOS. The study focuses mostly in which parts of the specification are more frequently included in actual implementations and how messages exchanged between clients and servers follow the structure defined by XML Schema files. The differences found between servers may shed some light to the cause of interoperability problems. The study may also show how different servers tend to group observations into offerings or which spatial features are more often used to represent the offerings, just to mention two possible outcomes.

The remainder of the paper is structured as follows. Section 2 introduces the SOS specification and lists the server instances used later on subsequent sections. After this, Section 3 presents the result of the analysis of the information gathered from the sever instances. In this section, we calculate the values of different metrics, such as number of invalid files, frequent validation errors, etc. Section 4 analyses which part of the schema files are used by the servers. Section 5 summarizes and discusses the results of the previous section. Lastly, we present the conclusions of our study.

2 Sensor Observation Services

The SOS specification provides a web service interface to retrieve sensor and observation data. The model used to represent the sensor observations defines the following concepts (OGC 2007, 2007b):

- *observation*: act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property;
- *feature of interest*: feature representing the real world object which is the observation target;
- *observed property*: phenomenon for which a value is measured or estimated;
- *procedure*: process used to produce the result. It is typically linked to a sensor or system of sensors; and
- *observation offering*: logical grouping of observations offered by a service that are related in some way.

The operations of the SOS specification are divided into three profiles (OGC 2007):

- *Core profile*: mandatory operations for any SOS server instance:
    - *GetCapabilities*: It retrieves metadata information about the service.
    - *DescribeSensor*: It retrieves information about a given procedure.
    - *GetObservation*: It retrieves a set of observations that can be filtered by a time instant or interval, location, etc.
- *Transactional profile*: optional operations for data producers to interact with the server:
    - *RegisterSensor*: It allows new sensors to be inserted.
    - *InsertObservation*: It allows new observations to be inserted.
- *Enhanced profile*: optional profile including a richer set of operations to interact with the server. For example:
    - *GetFeatureOfInterest*: Returns the geometry describing a feature of interest.
    - *GetResult*: It allows clients to reduce the transfer of redundant information related with sensor data when working with the same set of sensors.

The information about sensors and observations retrieved from the servers is usually encoded using SensorML (OGC 2007c) and O&M (OGC 2007b). Nevertheless, the specification allows data producers to encode data in their own favourite formats. SOS implementation also depends on other specifications such as Geography Markup Language (GML) (OGC 2004), OGC Web Services Common (OGC 2007d), and Filter encoding

specification (OGC 2005). All of these dependencies are shown in Figure 1.

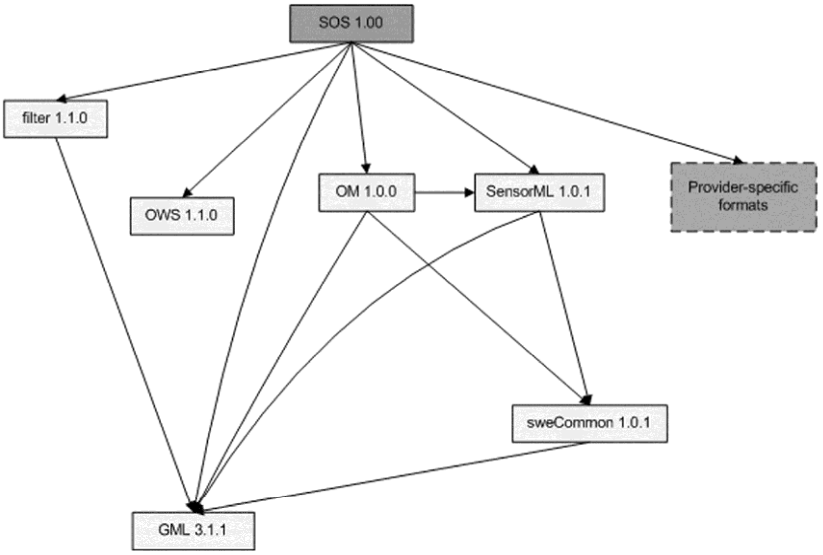


Fig 1: Dependencies of SOS from other specifications

2.1 SOS server instances

In order to realize our study, we gathered information from a set of SOS server instances freely available on the Internet. The URLs of these servers are listed in Table 1. These servers where located using web services catalogs, such as the OWS Search Engine[1] and IONIC RedSpider Catalog Client[2], and using general-purpose search engines such as Google and Yahoo!. The table only shows the servers claiming to support version 1.0.0 of the standard.

Starting from these servers we retrieved a sample set of XML instance files including service metadata and sensors and observations information. These instance files were then analysed mainly regarding to schema validity and used features. The results from this analysis are shown extensively in Section 3.

[1] http://ows-search-engine.appspot.com/index
[2] http://dev.ionicsoft.com:8082/ows4catalog/elements/sos.jsp

Table 1: List of SOS server instances

	Server URL
1	http://152.20.240.19/cgi-bin/oos/oostethys_sos.cgi
2	http://204.115.180.244/server.php
3	http://81.29.75.200:8080/oscar/sos
4	http://ak.aoos.org/ows/sos.php
5	http://bdesgraph.brgm.fr/swe-kit-service-ades-1.0.0/REST/sos
6	http://ccip.lat-lon.de/ccip-sos/services
7	http://compsdev1.marine.usf.edu/cgi-bin/sos/v1.0/oostethys_sos.cgi
8	http://coolcomms.mote.org/cgi-bin/sos/oostethys_sos.cgi
9	http://data.stccmop.org/ws/util/sos.py
10	http://devgeo.cciw.ca/cgi-bin/mapserv/sostest
11	http://elcano.dlsi.uji.es:8080/SOS_MCLIMATIC/sos
12	http://esonet.epsevg.upc.es:8080/oostethys/sos
13	http://gcoos.disl.org/cgi-bin/oostethys_sos.cgi
14	http://gcoos.rsmas.miami.edu/dp/sos_server.php
15	http://gcoos.rsmas.miami.edu/sos_server.php
16	http://gis.inescporto.pt/oostethys/sos
17	http://giv-sos.uni-muenster.de:8080/52nSOSv3/sos
18	http://habu.apl.washington.edu/cgi-bin/xan_oostethys_sos.cgi
19	http://lighthouse.tamucc.edu/sos/oostethys_sos.cgi
20	http://mmisw.org/oostethys/sos
21	http://nautilus.baruch.sc.edu/cgi-bin/sos/oostethys_sos.cgi
22	http://neptune.baruch.sc.edu/cgi-bin/oostethys_sos.cgi
23	http://oos.soest.hawaii.edu/oostethys/sos
24	http://opendap.co-ops.nos.noaa.gov/ioos-dif-sos/SOS
25	http://rtmm2.nsstc.nasa.gov/SOS/footprint
26	http://rtmm2.nsstc.nasa.gov/SOS/nadir
27	http://sccoos-obs0.ucsd.edu/sos/server.php
28	http://sdf.ndbc.noaa.gov/sos/server.php
29	http://sensor.compusult.net:8080/SOSWEB/GetCapabilitiesGFM
30	http://sensorweb.cse.unt.edu:8080/teo/sos
31	http://sensorweb.dlz-it-bvbs.bund.de/PegelOnlineSOS/sos
32	http://sos-ws.tamu.edu/tethys/tabs
33	http://swe.brgm.fr/constellation-envision/WS/sos-discovery
34	http://vast.uah.edu/ows-dev/dopplerSos
35	http://vast.uah.edu/ows-dev/tle
36	http://vast.uah.edu/vast/nadir
37	http://vast.uah.edu:8080/ows-dev/footprint
38	http://vastserver.nsstc.uah.edu/vast/adcp
39	http://vastserver.nsstc.uah.edu/vast/airdas
40	http://vastserver.nsstc.uah.edu/vast/weather
41	http://v-swe.uni-muenster.de:8080/WeatherSOS/sos
42	http://weather.lumcon.edu/sos/server.asp
43	http://webgis2.como.polimi.it:8080/52nSOSv3/sos

44	http://wron.net.au/BOM_SOS/sos
45	http://wron.net.au/CSIRO_SOS/sos
46	http://ws.sensordatabus.org/Ows/Swe.svc/
47	http://www.cengoos.org/cgi-bin/oostethys_sos.cgi
48	http://www.csiro.au/sensorweb/BOM_SOS/sos
49	http://www.csiro.au/sensorweb/CSIRO_SOS/sos
50	http://www.csiro.au/sensorweb/DPIW_SOS/sos
51	http://www.gomoos.org/cgi-bin/sos/V1.0/oostethys_sos.cgi
52	http://www.mmisw.org:9600/oostethys/sos
53	http://www.pegelonline.wsv.de/webservices/gis/sos
54	http://www.wavcis.lsu.edu/SOS/server.asp
55	http://www.weatherflow.com/sos/sos.pl
56	http://www3.gomoos.org:8080/oostethys/sos

2.2 Limitations of the Study

This study presents some limitations. First, it is impossible to retrieve all of the information published on the servers. We tried to overcome the effects of this limitation by making the sample dataset as large as possible and, in cases where several alternatives exists for making a request, we retrieved at least one instance file from each alternative. Second, only responses from the core profile operations were considered. This is because most servers do not implement the rest of the operations (see Section 3.1.2). Third, we did not test server instances for full compliance to the SOS specification; we only deal with the information contained in the XML instance files and XML schema files. Last, we analysed server instances without considering the server product used to deploy the instance. This is because for several instances we were not able to determine which product was used and in some cases handcrafted servers have been developed for specific problems.

2.3 Dataset Description

Details about the information contained in the sample dataset are presented in Table 2. The table includes the following information for the responses of the considered operations:

- *Number of files (NF):* Number of files retrieved for the operation.
- *Number of objects described (NO)*: Depending on the operations these objects are *observations offerings*, in the case of the *GetCapabilities* op-

eration; *sensor systems*, in the case of *DescribeSensor*; and *observations*, in the case of *GetObservation*.

Table 2: Dataset description

Operation	NF	NO
GetCapabilities	56	7190
DescribeSensor	6719	6719
GetObservation	204	3990656
Total	6979	4004565

3 Results

In this section we present the results of computing the sample dataset according to the following metrics:

- *Number of Invalid Files:* Number of files invalid according to the schema files.
- *Most frequent validation errors:* List with most frequent errors found during validation, including an error description and the number of occurrences of each error.
- *Used Features:* The features presented depend on the analysed operation. For example, while analysing capabilities files, we considered supported operations or filters and response formats. While analysing observation files, we considered, for example, which observation type is most frequently used to encode the information gathered by sensors.
- *Parts of the schemas that are actually used*: Schema files defining the message structures for SOS are large and complex, moreover, SOS schema files depend on schema files included on other specifications as well. For these reasons actual implementations only use a subset of these schemas.

We present the results of applying the first three metrics divided by operation. Then, in a different section, we analyse the part of schemas that are actually used.

3.1 Capabilities files

The capabilities file of a server contains all of the information needed to access the data it contains. In the case of SOS servers, this file contains available observation offerings, supported operations and filters, etc.

3.1.1 Instances validation

The first important fact extracted from the sample dataset is that 34 out of 56 (60.7%) capabilities files are *invalid* according to the schemas defining their structure. Table 3 shows the most frequent errors found in the instance files.

Table 3: Most frequent validation errors for capabilities files

	Error code	Description	Number of Occurrences
1	cvc-complex-type.2.4.a	Invalid content was found starting with element [element name]. One of {valid element list} is expected	2,754
2	cvc-complex-type.2.2	Element must have no element [children] and the value must be valid	978
3	cvc-datatype-valid.1.2.3	[value] is not a valid value of union type	960
4	cvc-attribute.3	The value of attribute on element is not valid with respect to its type	468
5	cvc-datatype-valid.1.2.1	[value] is not a valid value of union type	379
6	cvc-id.2	There are multiple occurrences of ID value	107

The most frequent error found was the use of a different name for an element than the one specified in the schemas. For example, this was the case for element *sos:Time*, which specifies the time instant or period for the observations within an offering. The element name was changed to *sos:eventTime* in some of the servers, maybe because that was the name in previous versions of the specification. The second most frequent error was elements with invalid content (errors 2, 3, 4 and 5). Common mistakes were time values with incorrect format or offering ID values containing whitespaces or colons.

Despite the large number of errors found, most of them did not prevent the files from being correctly parsed, although they supposed an extra amount of work while implementing the parsers. At the end, only 2 of the 56 files contained serious errors, which make parsing their content impossible for us.

3.1.2 Supported Operations

The capabilities files also indicate which operations are supported by the servers, including information about how to access them and which values are allowed as parameters. Table 4 shows which and how frequently the different operations are supported.

Table 4: Operations supported for the server instances

	Operation Name	Profile	GET Support	POST Support
1	GetCapabilities	Core	56	54
2	DescribeSensor	Core	33	45
3	GetObservation	Core	42	54
4	RegisterSensor	Transactional	0	2
5	InsertObservation	Transactional	1	2
6	GetFeatureOfInterest	Enhanced	0	12
7	GetObservationById	Enhanced	0	10
8	GetResult	Enhanced	0	1
9	GetFeatureOfInterestTime	Enhanced	0	0
10	DescribeFeatureType	Enhanced	0	0
11	DescribeObservationType	Enhanced	0	0
12	DescribeResultModel	Enhanced	0	0

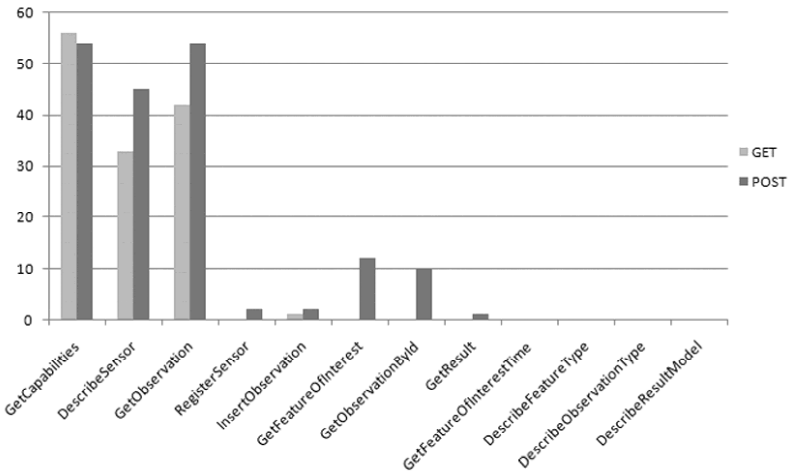


Fig 2: Support of SOS operations in actual server instances

The results, also depicted in Figure 2, show that all of the servers implement the *GetCapabilities* request using HTTP GET as required by the SOS implementation specification. Apart from that, most of them also implement the operation using HTTP POST. Most complex requests such as *GetObservation* are implemented easier using HTTP POST than using HTTP GET, as the SOS specification does not define KVP encodings for this operation.

The core profile is mandatory for every server, but 10 of the 56 servers do not implement the *DescribeSensor* request or at least they do not include it in the capabilities file. Operations for the transactional and enhanced profile are implemented by a few server instances and some of them are not implemented at all.

3.1.3 Supported Filters

The number of potential observations published on a server can be very large. For this reason, filters are used to request just the observations in which we are interested. Filters for SOS fall into four categories: spatial, temporal, scalar, and identifier filters. Only 16 of the 56 (28.5%) capabilities files include information about the supported filters. These filters are detailed in Table 5. For each filter category, the supported operands and operators are shown, as well as how frequently they have been used.

The most implemented filters are *BBOX* and *TM_During* that allow restricting the location of the observations to a given bounding box or to a given time period, respectively. *Id filters* are also frequently implemented. They allow information to be filtered by specifying the ID of entities related with the request. Even though some servers do not include the filter capabilities section, most of them allow observations to be filtered using a bounding box or a time interval.

3.1.4 Supported Response Formats

Observations published on different server instances are encoded using several different formats. These formats and the number of offerings that represent information with them are presented in Table 6. The most supported format to represent observations is O&M 1.0.0, which is the default format specified by SOS. A deeper discussion about this format is presented in Section 3.3.

Table 5: Support of filters

Filter Category			Number of Appearances
Spatial Filters	*Operands*	gml:Envelope	16
		gmlPolygon	11
		gml:Point	11
		gml:LineString	11
	Operators	BBOX	15
		Contains	11
		Intersects	11
		Overlaps	11
		Equals	1
		Disjoint	1
		Touches	1
		Within	1
		Crosses	1
		DWithin	1
		Beyond	1
Temporal Filters	*Operands*	gml:TimeInstant	16
		gml:TimePeriod	16
	Operators	TM_During	15
		TM_Equals	14
		TM_After	14
		TM_Before	14
		TM_Begins	1
		TM_Ends	1
Scalar Filters	*Operators*	Between	14
		EqualTo	13
		NotEqualTo	13
		LessThan	13
		LessThanEqualTo	13
		GreaterThan	13
		GreaterThanEqualTo	13
		Like	12
		NullCheck	1
Id Filters		eID	16
		fID	15

Table 6: Formats supported to represent observation information

Format	Number
Text/xml; subtype="om/1.0.0"	5110
Text/xml;schema="ioos/0.6.1"	2064
Text/csv	664
application/vnd.google-earth.kml+xml	664
Text/tab-separated-values	664
application/zip	110
Text/xml	4
application/com-binary-base64	1
application/com-tml	1

3.1.5 Offerings Information

Observation offerings contain information about a set of related sensor observations. The SOS specification does not say how observations, procedures or observed properties should be grouped into offerings. For this reason, it would be very interesting to know how this grouping is realised in actual implementations. Regarding observation offerings we computed the following metrics:

- *Number of offerings per server (OpS)*: How many offerings are usually published on a server;
- *Number of procedures per server (PpS)*: How many sensor or sensor systems are published on a server;
- *Number of observed properties per server (OPpS)*: How many observed properties are usually published on a server; and
- *Number of offering as points*: An interesting peculiarity observed during the experiments is that the location of most offerings is a point, instead of a bounding box.

The result of computing the first three metric values is shown in Figure 3. The figure shows the values grouped into 6 categories. The number of offerings per server ranges from 1 to 1772. 48% of the servers contain 1-4 offerings and 63 % contain 16 or less. This indicates that servers tend to group observations in a few offerings. Similarly, the number of procedures per server ranges from 1 to 1957. Although in a lesser degree than the case of offerings, the number of servers with a large amount of procedures per server is always lower than the number of servers with a small number of procedures. The number of observed properties per server ranges from 1 to 114. This number behaves much like the previous ones having 65% of the server instance with less than 16 observed properties advertised.

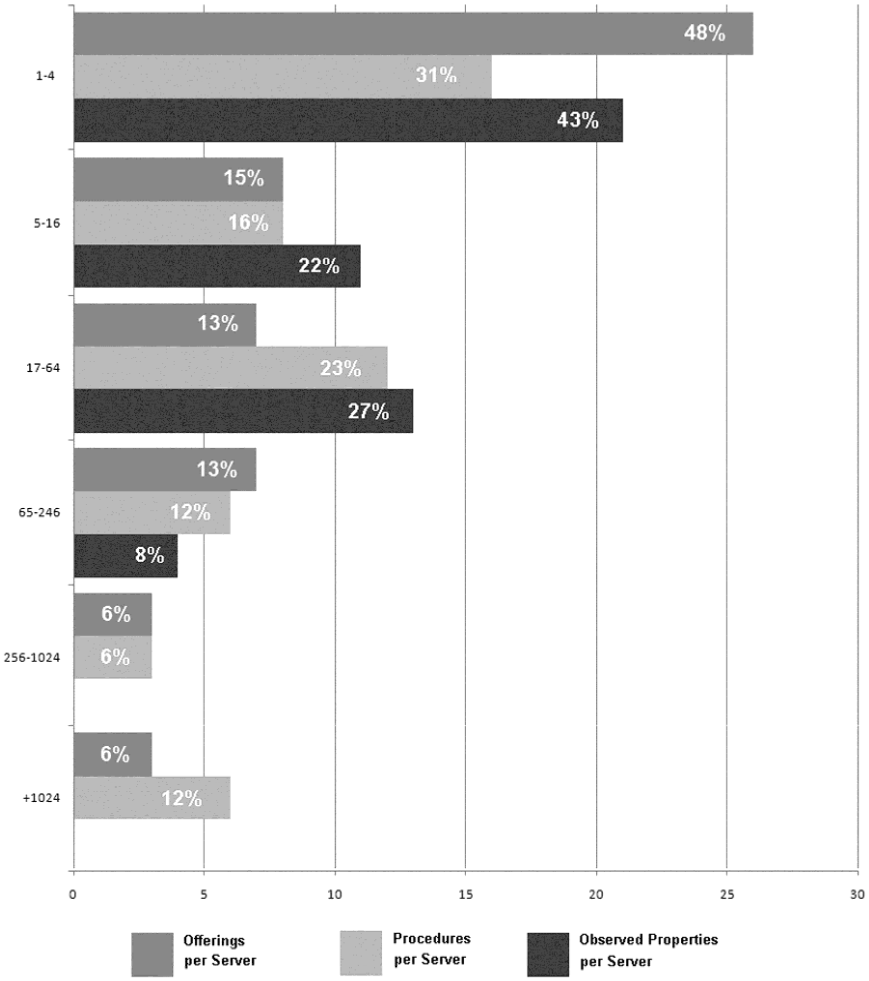


Fig 3: Number of servers classified for the number of offerings, procedures, and observed properties that they contain

A last interesting phenomenon found here is the number of observation offerings which are restricted to a point in the space. Each offering has a property named *boundedBy* defining a bounding box where the observations grouped in the offering are located. In 6575 offerings in the sample data set, this bounding box was indeed a point, representing 95.7% of the total number of offerings. This clearly indicates that the first criteria followed to group observations into offerings is the sensors location, which in most of the cases is a single point on the Earth. Figure 4 shows as an example a set of offerings located in North America represented in Google

Earth. In the figure, *placemarks* represent *point offerings* and bounding boxes represent other offerings.

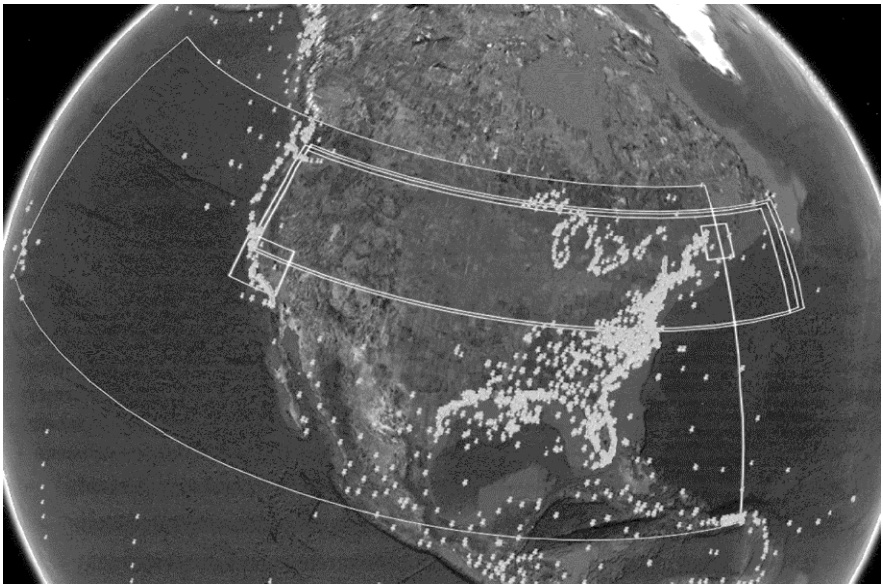


Fig 4: Observation offerings in North America

3.2 Procedure description files

The 56 servers considered in this study mention in their capabilities files 12,222 procedures. From this number we were able to retrieve the description files of 6719 of them (54.9%). All of these files were encoded in the sensorML format.

3.2.1 Instances validation

The validation of the sensorML files gave as a result that 1896 of the files were invalid according to the XML schemas files defining the structure of these documents. The value represents 28.2% of the overall number of files. The most frequent errors found are presented in Table 7.

The first error type occurred frequently because required elements were omitted or elements not defined in the schemas were introduced in the wrong place. Errors 2, 4, and 5, similar to the case of the capabilities files, refer to incorrect formatted values: identifiers including whitespaces or colons, incorrect time values, or values just being left empty. The most seri-

ous errors were those of type 2. In these cases, wrong use of namespaces or not specifying the version of the schemas used made it impossible to process the documents at all.

Table 7: Most frequent validation errors for sensor description files

	Error code	Error Description	Number of Occurrences
1	cvc-complex-type.2.4.a	Invalid content was found starting with element [element name]. One of {valid element list} is expected	1778
2	cvc-attribute.3	The value of attribute on element is not valid with respect to its type	556
3	cvc-elt.1.a	Cannot find the declaration of element [element name]	500
4	cvc-datatype-valid.1.2.1	[value] is not a valid value of union type	300
5	cvc-pattern-valid	Value is not facet-valid with respect to pattern for type	256

3.2.1 Procedure description types

The sensorML specification models sensor systems as a collection of physical and non-physical processes. Physical processes are those where information regarding their positions and interfaces may be relevant. Examples of these processes are detectors, actuators, and sensor systems. Non-physical or "pure" processes according to the specification "*can be treated as merely mathematical operations*" (OGC 2007c). These categories are further subdivided as shown next:

- *Physical processes*
  - *Component*: Any physical process that cannot be subdivided into smaller subprocesses.
  - *System*: It may group several physical or non-physical processes.
- *Non-physical processes*
  - *Process Model*: Defines an atomic pure process which is used to form process chains.
  - *Process Chains*: Collection of executable processes in a sequential manner to obtain a desired result.

From 6219 processed sensorML files, 6215 described *Systems* (99.9%) and 4 of them described *ProcessChains*. This indicates that the usual is to describe sensor systems that have a location in space and measure an observed property for a period of time.

3.2.1 Specifying location

An important piece of information about the procedure is its *location*. Unfortunately for programmers, location can be specified in different parts of the procedure description file (sensorML file). In the sample dataset, we have found this information located in at least three different places and using different names to identify coordinates:

- Under the *location* tag in the description of a *System* as a point:

```
<SensorML xmlns="http://www.opengis.net/sensorML/1.0.1"
  version="1.0.1" [Other attributtes]>
  <member>
    <System gml:id=[System ID]>
      ...
      <location>
        <gml:Point srsName=[SRS Name]>
          <gml:coordinates>39.99 -0.068 0</gml:coordinates>
        </gml:Point>
      </location>
      ...
    </System>
  </member>
</SensorML>
```

- Under the *position* tag in the description of a *System* as a vector with named elements:

```
<SensorML xmlns="http://www.opengis.net/sensorML/1.0.1"
  version="1.0.1" [Other attributtes]>
  <member>
    <System gml:id=[System ID]>
      ...
      <sml:position name=[name]>
        <swe:Position referenceFrame=[SRS name]>
          <swe:location>
            <swe:Vector>
              <swe:coordinate name="x">
                <swe:Quantity>
                  <swe:value>-0.068</swe:value>
                </swe:Quantity>
              </swe:coordinate>
              <swe:coordinate name="y">
                <swe:Quantity>
                  <swe:value>39.99</swe:value>
                </swe:Quantity>
```

```
              </swe:coordinate>
              <swe:coordinate name="z">
                <swe:Quantity>
                  <swe:value>0</swe:value>
                </swe:Quantity>
              </swe:coordinate>
            </swe:Vector>
          </swe:location>
        </swe:Position>
      </sml:position>
      ...
    </System>
  </member>
</SensorML>
```

- Under the *positions* tag in the description of a *System* as a list of positions:

```
<SensorML xmlns="http://www.opengis.net/sensorML/1.0.1"
  version="1.0.1" [Other attributtes]>
  <member>
    <System gml:id=[System ID]>
      ...
      <sml:positions>
        <sml:PositionList>
          <sml:position name=[name position 1]>
            [Position data]
          </sml:position>
          <sml:position name=[name position 2]>
            [Position data]
          </sml:position>
          ...
        </sml:PositionList>
      </sml:positions>
      ...
    </System>
  </member>
</SensorML>
```

In the first case reading, the coordinate values are straightforward; the values are grouped together into a *gml:Point* object. In the second one, several tags must be parsed to reach the coordinates; a problematic issue at this point is that different names are used by servers to refer to the coordinate values. For example, *longitude* was also named *x* or *easting*; *latitude* was also named *y* or *northing*; and *altitude* was also named *z*. The contents and attributes of the tags involved are also slightly different. Some servers include unit of measurements; some include the axis they refer to, etc. The third case is a generalization of the second one, where positions are included in a list, allowing more than one to be specified. None of the analysed files included more than one position for a sensor or sensor system.

3.3 Observation files

To analyse *GetObservation* responses, 1.7 GB of observation data were retrieved from the server instances. All of the retrieved files follow the format specified by O&M 1.0.0 encoding specification. As shown in Table 6, this is the most widely used format and is the default for encoding observations in SOS 1.0.0.

3.3.1 Instances validation

Validation of observation files was much more difficult than expected. The validation process repeatedly failed to correctly process large files (> 10MB) and did not allow the validation of files containing *measurements* alleging that schema files were incorrect. *Measurements* are specialized observations where the observation value is described using a numeric amount with a scale or using a scalar reference system (OGC 2007b). Large files were only a few, so the first limitation was not a great problem but files containing measurements were about half of the whole observation files. Although we were able to parse correctly all of the observations, we were only able to apply the validation process to 62 files (31.3%). From these 62 files, 56 were reported to be invalid (90%). Details about the errors found are shown in Table 8.

Table 8: Most frequent validation errors for sensor description files

	Error code	Error Description	Number of Occurrences
1	cvc-attribute.3	The value of attribute on element is not valid with respect to its type	206
2	cvc-complex-type.2.4.a	Invalid content was found starting with element [element name]. One of {valid element list} is expected	189
3	cvc-datatype-valid.1.2.1	[value] is not a valid value of union type	121

The validation errors for observation files are similar to those for capabilities and sensor description files. Values with wrong formats and wrongly named or misplaced elements made up all of the errors found in the instance files.

3.3.2 Observation Types

According to the O&M 1.0.0, encoding specification observation types are organized as shown in Figure 5. The base type for all observations is *ObservationType*, which inherits form *AbstractFeatureType* located in GML schemas. Starting from *ObservationType,* a set of specializations is defined based on the type of the results contained in the observations. Additionally, information providers can derive their own observation data types from the different types in the figure.

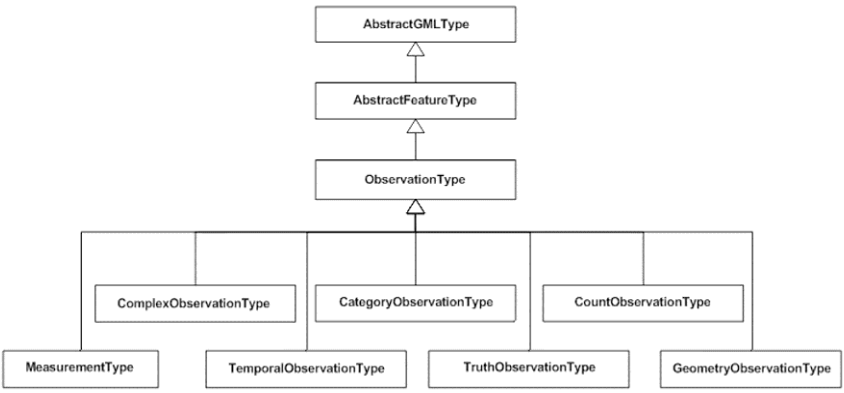


Fig 5: Hierarchy of observation types

From 3,990,656 observation values processed in the dataset, 56.3 % (2,246,639) of the values were *Observation* elements (instances of *ObservationType)* and 43.7 % (1,744,017) were instances of *Measurement* elements (instances of *MeasurementType*). Values corresponding to none of the other types were found in the sample dataset. Despite the fact that the number of measurement values was lower than the number of observations, the amount of disk space needed to contain these values was about 7 times larger than the space occupied by the observations (1533 MB against 213 MB). This difference in size seems to be the reason why most implementations choose not to use observation specialization types, although the lack in the O&M specification of well-defined semantic models might influence this decision as well (Probst 2008, Kuhn 2009).

3.4 Subset of XML Schemas used

The last piece of information we extracted from the sample dataset is the subset of the XML Schemas that is actually used in the server instances. The number of schema files associated to the SOS specification is huge. If we follow all of the dependencies from the main schema files of the specification, we obtain a set of 87 files. If we additionally consider the observations specialization schemas (containing the definition of *MeasurementType*) and their own dependencies, this number grows to 93. The size of schemas brings as a consequence that server instances only provide support for a subset of them.

Next, we calculate from the sample dataset which part of the schemas is used and which part is not used at all. To calculate this information we inspect the information contained on the instance files to determine which schema components are directly used in the files (*initial set*). After doing this, we determine which other schema components are used to define the initial set. The algorithm used is similar to the one included in the GML subsetting profile tool, a tool used to extract subsets of the GML schemas (OGC 2004). We present the results in two steps. Firstly, we detail the subset of the GML schemas that is actually used and secondly, a similar analysis with the overall results for the SOS specification is presented.

3.4.1 GML

GML constitutes more than 50% of the overall number of global schema components (types, elements, model groups) comprising the SOS schemas. It is used to model geographic features embedded into the instance files and its components are extended or composed into new components of the SOS specification. As shown in Figure 1, most of the specifications relevant to our study depend to a large extent on GML.

Table 9 shows a comparison between the number of components in the original GML schemas for version 3.1.1 (*original files*) and the subset of the schemas that is referenced directly or used in the definition of other components referenced directly in the sample dataset (*profile*).

The results are divided by component type: complex types (#CT), simple types (#ST), global elements (#EL), global attributes (#AT), model groups (#MG), and attribute groups (#AG). It turned out that only 16.3% of the components were actually used. All of the components contained in the following files were not used at all: *coverage.xsd, dataQuality.xsd, defaultStyle.xsd, direction.xsd, dynamicFeature.xsd, geometricComplexes.xsd, geometricPrimitives.xsd, grids.xsd, measures.xsd, temporal-*

referenceSystems.xsd, temporalTopology.xsd, topology.xsd, and *valueObjects.xsd*.

Table 9: Comparison between overall number of components and number of components actually referenced in GML

	Original files	Profile
#CT	394	60
#ST	64	15
#EL	485	74
#AT	15	9
#MG	12	2
#AG	35	4
Total	1005	164

3.4.2 SOS

As mentioned before, the full SOS schemas are comprised of 93 files, distributed by specification as presented in Table 10. This full set is calculated starting from the SOS *"main schemas"* and following the references specified with *include* and *import* tags. For example, a typical practice when accessing a component in the GML schemas is to import the whole schemas through the file *gml.xsd*. This way all of the GML schemas become referenced even when most of them are never used.

Table 10: Distribution of SOS 1.0.0 schema files by specification

Specification	Version	Number of files
SOS	1.0.0	16
GML	3.1.1	32
SensorML	1.0.1	5
OM	1.0.0	3
SWE Common	1.0.1	11
Sampling	1.0.0	5
OWS	1.1.0	14
Filter	1.1.0	4
Others		3

Table 11 shows a comparison between overall number of components in the full SOS schemas (*original files*) and the subset of the components that is really needed as explained before in the case of GML (*profile*). The results are also displayed in Figure 6.

Table 11: Comparison between overall number of components and number of components actually used in the SOS full schema set

Metric	Original files	Profile
#CT	772	266
#ST	119	61
#EL	745	201
#AT	39	3
#MG	28	16
#AG	40	8
Total	1743	515

Only 29.5% of the components in the full schema set are actually used in the sample dataset.

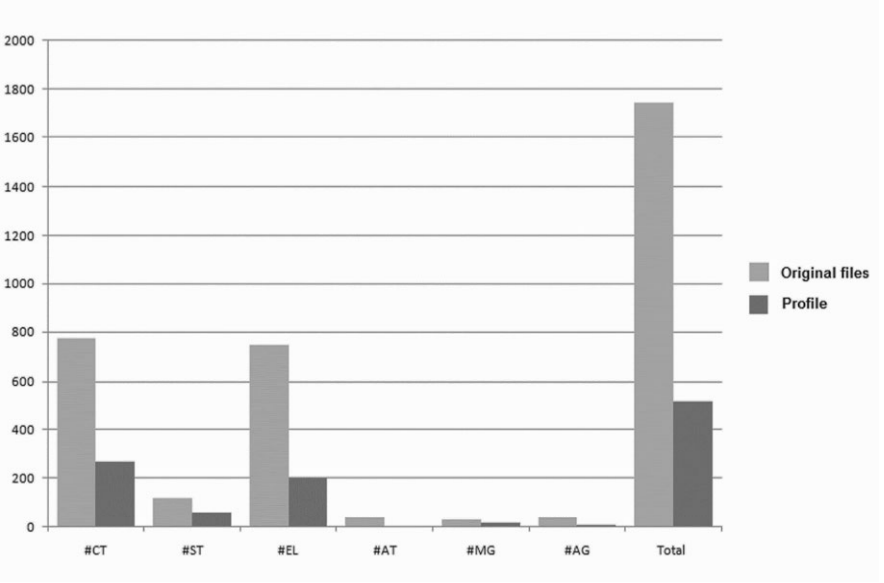


Fig 6: Overall number of schema components vs actually used components in SOS.

4 Discussion

As the amount of information extracted from the sample dataset is large we present a summary of our findings:

1. The number of invalid instances files is high: 29% (1986 out of 6837);
2. Most of the validation errors found are not serious enough to prevent correct parsing in many cases;
3. Some servers do not implement all of the mandatory operations in the core profile;
4. Most servers do not advertise filtering capabilities;
5. Most servers use O&M to encode observations;
6. Most servers group observations into a small number of offerings and they usually contain information about a small number of procedures and observation properties;
7. Offerings location is frequently a point in space indicating that the first criteria to distribute observations into offerings is the sensors location;
8. Most procedure descriptions refer to *Systems*;
9. All of the observations in the sample dataset belong to only two types: *observations* and *measurements*;
10. The size on disk needed to represent *measurements* is a lot higher than to represent the same information as basic *observations*;
11. Most servers only support operations from the core profile;
12. Procedure location is specified in at least three different parts of the sensorML documents and sometimes coordinates are referred to under different names. This problem could be solved by allowing only one of the three choices. If multiple locations can be specified for a procedure, the more general solution would be the most appropriate, although we did not find any instance with more than one location in the sample dataset; and
13. Only 29.5 % of the full schema set for SOS is used by the sample dataset.

The first four points are closely related to interoperability. The presence of invalid files increases the chance of parsing errors in client-side applications. The fact that most errors are easy to overcome if writing the parsers manually does not deny the fact that the errors may limit the applicability of XML data binding code generators if they are strict regarding schemas validity. Not supporting mandatory operations may also lead a client to fail if they request these operations to the server. Not advertising filtering capabilities simply prevents the clients to effectively filter the observations, unless they know beforehand how the server works.

The next six points (5-10) provide useful insight for optimizing server and client implementations. Offering grouping strategies and knowing which formats and types of sensor and observation representations that are more commonly used could optimize implementations to these scenarios.

Even more, they could indicate which features are most likely to stay in future versions of the specification. Point 10 is especially revealing if large amounts of information are being handled. In this case, using measurements are not the right choice for encoding information.

The last three points, in our opinion, reflect the complexity of the SOS specification. The number of operations in the specification is high if compared with other OGC specifications. In addition, the complexity of the formats that must be supported such as SensorML, O&M, SWE common, and GML, makes the implementation of the core profile itself a complex task. The example of how location is specified for procedures shows that even getting a simple piece of information can be a difficult thing to do. The last point could be the result of two options: the schemas are too complex to be implemented in its entirety or most of the information included or referenced by the schemas is not needed in real scenarios. In our opinion both options are true to a certain degree. Schemas are complex enough to make it almost impossible to fully implement them manually. This complexity also makes code generation based on them tricky, as they use schema features that are not supported by some generators. In addition, some schemas contain validation errors. Regarding if all of the information included in the schemas are really needed, they have been designed to be useful in as many scenarios as possible. Even if the design process starts with a very well defined use case, how real users are going to utilise them is not easy to predict.

5 Conclusions

In this paper, we have presented an empirical study of actual instances of servers implementing SOS. The study focused mostly in which parts of the specification are more frequently included in real implementations and how exchanged messages follows the structure defined by XML Schema files. Several interesting outcomes have been obtained, such as the main criteria to group observations into offerings, the small subset of the schemas that are actually used, the large number of files that are invalid according to the schemas, etc.

All of these findings must be taken with care because the study has presented several limitations, such as the impossibility to retrieve all of the information published on servers or only the responses from the core profile operations were considered. Nevertheless, they can be of practical use when implementing SOS servers and clients. For example, to decide which parts of the schemas to support, to suggest how to encode large datasets of

observations, to know where to look for the sensors' location, just to mention some. As future work, we are trying to use all of this information to build customized SOS servers and clients that allow large amounts of data to be handled efficiently.

References

Botts M, Percivall G, Reed C, Davidson J (2008) OGC® Sensor Web Enablement: Overview and High Level Architecture. In: GeoSensor Networks, LNCS, 4540/2008, 175-190.

Kuhn W (2009) A Functional Ontology of Observation and Measurement. In: Proceedings of the 3rd International Conference on GeoSpatial Semantics (GeoS '09), 26-43.

OGC (2004) OpenGIS® Geography Markup Language (GML) Implementation Specification 3.1.1.OGC Doc. Number 03-105r1

OGC (2005) OpenGIS® Filter Encoding Implementation Specification 1.1.0. OGC Doc. Number 04-095

OGC (2007) Sensor Observation Service 1.0.0. OGC Doc. Number 06-009r6

OGC (2007a) OpenGIS® Sensor Planning Service Implementation Specification 1.0.0 OGC Doc. Number 07-014r3

OGC (2007b) Observations and Measurements – Part 1 - Observation schema. 1.0.0. OGC Document Number 07-022r1

OGC (2007c) OpenGIS® Sensor Model Language (SensorML) Implementation Specification. Version. 1.0.1. OGC Document Number 07-000

OGC (2007d) OGC Web Services Common SpecificationVersion. 1.1.0. OGC Document Number 06-121r3

Probst F (2008) Observations, measurements and semantic reference spaces. Applied Ontology - Ontological Foundations of Conceptual Modelling, 3(1-2): 63-69

van Zyl TL, Simonis I, McFerren G (2009) The Sensor Web: Systems of sensor systems. International Journal of Digital Earth 2(1): 16–30

W3C (2004) XML Schema Part 1: Structures Second Ed., http://www.w3.org/-TR/xmlschema-1. Last accessed 2010-09-21

W3C (2004a) XML Schema Part 2: Datatypes Second Ed., http://www.w3.org/-TR/xmlschema-2. Last accessed 2010-09-21

Qualitative Spatio-temporal Reasoning from Mobile Sensor Data using Qualitative Trigonometry

Juliane Brink

Institute for Geoinformatics, University of Muenster, Muenster, Germany
juliane.brink@uni-muenster.de

Abstract. This paper presents a method for qualitative spatio-temporal reasoning about dynamic spatial regions from mobile sensor data based on qualitative trigonometry. We apply this method to the use case of monitoring a travelling gas plume with a mobile sensor. We argue that our method can infer qualitative information about size, movement direction, and speed of a spatial region from the data of a mobile sensor passing through it, which allows for adapting the route of the sensor that captures the phenomenon in space and time.

1 Introduction

The volcanic ash plumes over Europe created by the eruptions of the Eyjafjallajökull in Iceland in 2010 and the underwater oil plumes caused by the Deepwater Horizon Oil Spill disaster in the Gulf of Mexico are two recent examples of dynamic environmental phenomena that are challenging to measure and delineate. Three-dimensional ash and oil concentrations are highly dynamic, invisible phenomena, and there is little sensing infrastructure available in place. The same applies to the dispersion of gas or particles after some emission in an accident. Particularly in those cases where the affected area is not accessible by humans, the obvious solution is to send one or more unmanned mobile vehicles equipped with sensors to autonomously perform a surveillance task. However, such a mobile agent

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_11,

needs to know how to interpret the spatio-temporal sensor data to derive an intelligent movement strategy for efficient data collection.
When we use mobile sensors for example carried by autonomous helicopters for observing spatio-temporal phenomena like a travelling gas plume we obtain data streams associated with trajectories in space and time. This raises the question whether variations in the sensor data stem from some variation or movement of the phenomenon over time, from the movement of the sensor, or from combinations of these. The sensor data stream depends on the speed and sampling rate of the sensor, the spatio-temporal properties of the observed phenomenon, and essentially the geometric configurations of the phenomenon and the sensor during the observation. Some of these parameters may be known such as the sensor characteristics, while others may be known with some imprecision and some may not be known at all.

All this information is crucial if we want to adapt the sensor movement to efficiently monitor a dynamic spatial phenomenon. Existing work in sensor motion planning addresses the problem using geostatistical methods, e.g. kriging (Walkowski 2008) or parameter estimation of distributed systems (Pantan and Uciski 2004; Song, et al. 2007). These approaches use a quantitative, field representation of the phenomenon.

However, research in Artificial Intelligence supports the view that qualitative representations and reasoning is an effective approach for robot exploration, mapping, and navigation in space (Cohn and Renz 2008). In qualitative spatial reasoning, a situation is characterized by categorical variables, which can take only a small number of values and can be used by inference rules. Unlike quantitative methods, qualitative spatial reasoning is possible with incomplete and imprecise information (Frank 1996). It also yields qualitative, i.e. less precise results, which might be desirable, if the precision is just sufficient for a robot to perform a planning task. Qualitative information is ‘cheaper’ than quantitative information since it is less informative (Freksa 1992). Thus, qualitative reasoning is a reasonable approach for intelligent autonomous agents operating in dynamic environments as it fits the limited resources of memory and computing time (Dutta 1990).

Qualitative approaches can also be found in the field of Geosensor Networks. Complex geographical phenomena are represented using qualitative models focusing on spatial entities like regions and boundaries (Duckham et al. 2005; Worboys and Duckham 2006; Jiang and Worboys 2008; Shi and Winter 2010; Duckham et al. 2010).

This paper will elaborate on the use of a moving sensor for observing a dynamic spatial region. Spatial regions are extracted out of continuous space having definable but non-existent boundaries and can be conceptual-

ized as objects (Bian 2007). In the following, we will use the term object in this sense to refer to spatial regions that can be travelled through and observed by a sensor. We analyse the interrelationships between the movement direction and speed of the sensor, the movement direction and speed of the observed object, and the size of the object. Based on this analysis, we propose a method for reasoning about the qualitative characteristics, speed, direction, and size of the object from incomplete and imprecise information about these characteristics and the measurements of a sensor moving through the object. To the best of our knowledge, this problem has not been addressed using qualitative methods so far. We apply a method of spatial reasoning about distances and angles based on qualitative trigonometry (Liu 1998). The motivating problem for this research is the conceptual development of a mobile sensor that is able to autonomously monitor moving objects in space and time by adapting its route based on qualitative spatio-temporal reasoning from the sensor data. We argue that this task can be accomplished using the qualitative and thus imprecise information that our approach is able to infer from the sensor data stream. Our method is applied to the use case of a travelling gas plume.

The remainder of this paper is structured as follows. In Section 2, we describe the geometric configuration of a sensor travelling through a moving object over time and analyse the distances and angles involved. We develop a method for spatio-temporal reasoning from sensor data using qualitative trigonometry. In Section 3, the use case of monitoring a travelling plume is described and our method is demonstrated using simulated data of a gas plume. Finally in Section 4, we point out some findings from this research, which are relevant for sensor movement planning for efficient monitoring of moving objects based on our method and discuss some directions for future work. Section 5 gives the conclusion.

2 Reasoning about moving objects from mobile sensor data

2.1 Sensor movement and object movement

Let us assume a sensor travelling at constant speed on a straight line in space. We will first assume that the sensor can only distinguish between presence and absence of a property, which corresponds to being inside or outside an observed object. In Section 3.2, we will relax this assumption.

Further, let us assume that for the duration of the sensor observation the object to be observed travels at constant speed on a straight line in space. If the sensor meets the object in space and time, there will be some point where the sensor enters the object $P_e = (x_e, y_e, z_e, t_e)$ and some point where it leaves the object $P_l = (x_l, y_l, z_l, t_l)$. During the interval $[t_e, t_l]$ the sensor has travelled some distance and the object has travelled some distance. Thus, the size of the object along the line travelled by the sensor does not correspond to the distance travelled by the sensor, but rather to a combination of the movement of the sensor and the movement of the object. This becomes particularly obvious if we look at the case where the sensor moves into the opposite direction of the object. Here the size of the object is bigger than the distance travelled by the sensor between entering and leaving the object. Another simple observation is that the direction in which a sensor passes an object is a combination of the travelling directions of the sensor and of the object as Figure 1 illustrates. If, for example, the object is moving from east to west and the sensor is approaching the object from the south at roughly the same speed, then relative to the object the sensor passes the object in north-easterly direction.

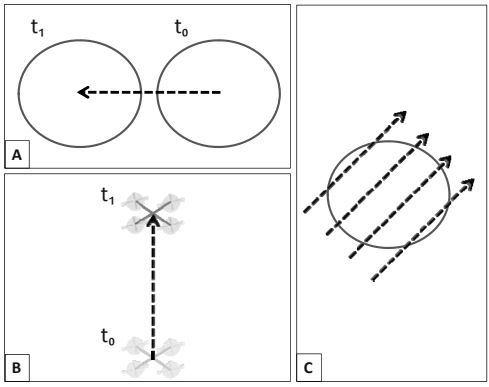


Fig. 1. If the object is moving from east to west (A) and the sensor is approaching the plume from the south to the north (at roughly the same speed) (B), then relative to the object the sensor passes the object in north-easterly direction (C).

Taking a closer look at this geometric configuration reveals that the distance travelled by the sensor, the distance travelled by the object, and the length of object passed by the sensor form a triangle (see Figure 2) that defines the angle between sensor and object movement directions. Thus, the ΔABC contains information about four fundamental properties of the observation process and the observed moving object, of which the fourth can be inferred, if any three of the four properties are known:

- the distance travelled by the sensor *AB*, which is known if we assume that the speed of the sensor is controllable,
- the distance travelled by the object *AC*,
- The size of the object along the line upon which it was passed by the sensor *BC*, and
- the angle between the travelling direction of the sensor and the travelling direction of the object α.

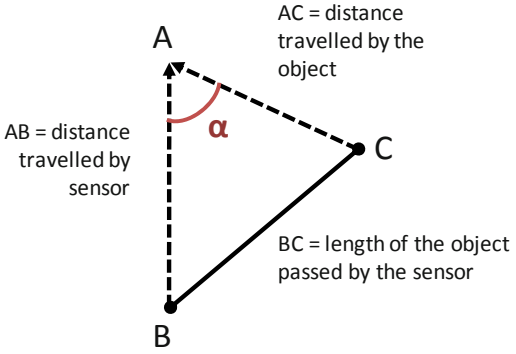


Fig. 2. The triangle of the distance travelled by the sensor, the distance travelled by the object, and the size of the object along the line passed by the sensor define the angle between sensor and object movements.

2.2 Qualitative trigonometry

In practical situations there is often some information available about the object properties size, speed, and movement direction. However, this information may be incomplete and imprecise. Thus, we propose a qualitative approach, limiting the possible distances and angles to sets of qualitative categories. This also allows reasoning about the unknown properties with less than three known properties by reasoning from a limited amount of possible combinations of categories. Our method uses qualitative trigonometry for inferring the properties of the triangle described above. We employ the definition of the semantics of qualitative distance, qualitative angles, and the qualitative trigonometric inference rules by Liu (1998), who defines qualitative distance values *x* relative to a reference constant *d* as

$$Less = \{x \mid x \in \Re, 0 < \frac{x}{d} \leq \frac{2}{3}\} \tag{1}$$

$$SlightlyLess = \{x \mid x \in \Re, \frac{2}{3} \leq \frac{x}{d} < 1\} \tag{2}$$

$$Equal = \{x \mid x \in \Re, \frac{x}{d} = 1\} \tag{3}$$

$$SlightlyGreater = \{x \mid x \in \Re, 1 < \frac{x}{d} \leq \frac{3}{2}\} \tag{4}$$

$$Greater = \{x \mid x \in \Re, \frac{3}{2} < \frac{x}{d}\} \tag{5}$$

In the following the qualitative values *Less*, *SlightlyLess*, *Equal*, *SlightlyGreater*, and *Greater* will also be denoted as *l*, *sl*, *eq*, *sg*, and *g*, respectively. The semantics of the qualitative values for angles are defined in (Liu 1998) as

$$Acute = \{\theta \mid 0 \leq \theta < \frac{\pi}{3}\} \tag{6}$$

$$SlightlyAcute = \{\theta \mid \frac{\pi}{3} \leq \theta < \frac{\pi}{2}\} \tag{7}$$

$$RightAngle = \{\theta \mid \theta = \frac{\pi}{2}\} \tag{8}$$

$$SlightlyObtuse = \{\theta \mid \frac{\pi}{2} < \theta \leq \frac{2\pi}{3}\} \tag{9}$$

$$Obtuse = \{\theta \mid \frac{2\pi}{3} < \theta \leq \pi\} \tag{10}$$

In the following the labels *Acute*, *SlightlyAcute*, *RightAngle*, *SlightlyObtuse*, and *Obtuse* will also be denoted as *a*, *sa*, *r*, *so*, and *o*, respectively.

Based on these definitions qualitative trigonometric inference rules can be formulated that describe the relationships of distances and angles in a triangle. The detailed formalism can be found in Liu (1998).

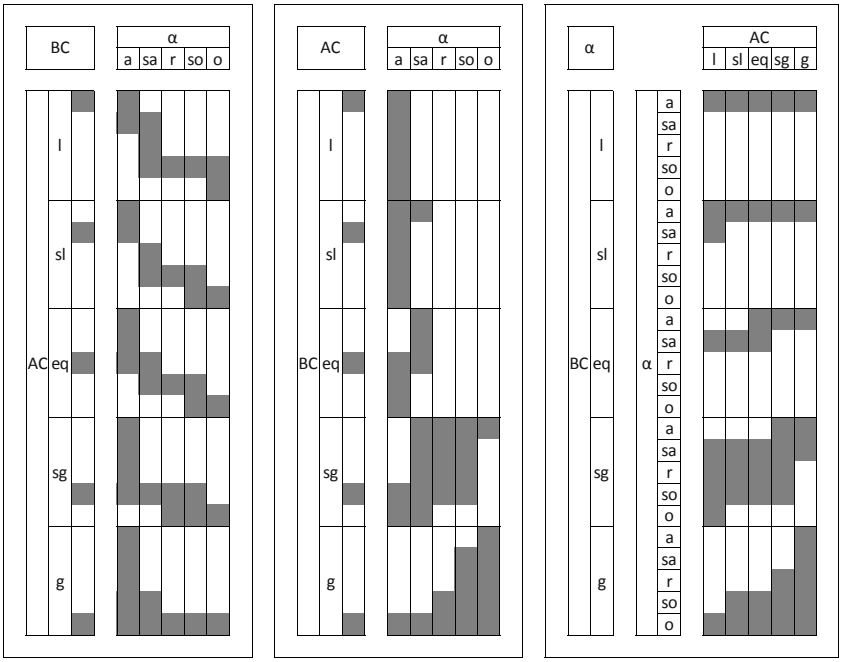


Fig. 3. The tables show the inference rules for reasoning about size and movement of objects from mobile sensor data using qualitative trigonometry (partially adapted from Liu, 1998))

We adapt this formalism to handle our problem of reasoning about moving objects from mobile sensors. Reconsider the triangle depicted in Figure 2. If we choose the reference distance $d = AB$ as the distance travelled by the sensor, we can formulate qualitative inference rules for BC, AC, and α as depicted in Figure 3. Using these rules it is possible to reason about the size and the speed of the object and the angle between movement direction of the sensor and movement direction of the object, i.e. the absolute movement direction of the object, based on the sensor observation and incomplete and imprecise information about object properties. The first table in Figure 3 contains the inference rules for reasoning about BC (size of the object along the line travelled by the sensor) from the qualitative values of AC (distance travelled by the object) and α (angle between sensor and object movement). The second and the third table contain the inference rules for reasoning about AC from the qualitative values of BC and α and about α from the qualitative values of BC and AC, respectively. Thus, computing the qualitative value of an unknown property is a simple table look up using qualitative information about the two other properties. For example, if the distance travelled by the object AC is *SlightlyLess* than the distance

travelled by the sensor and the angle between sensor and object movement α is *SlightlyAcute*, using the first table you can infer that the size of the object (along the line travelled by the sensor) *BC* is *Equal* to *SlightlyGreater* than the distance travelled by the sensor.

Note that with many combinations of property values reasoning is only possible with some ambiguity. However, due to the limited combinations of qualitative values, reasoning is also possible with multiple possible qualitative values for an input property or with an input property being completely unspecified.

3 Use case: Monitoring a moving gas plume

3.1 Gaussian plume model

We want to demonstrate the applicability of the described qualitative spatio-temporal reasoning approach looking at a specific use case. A large share of spatio-temporal phenomena can be modelled as a plume that is moving in space. Any puff emission of gas or particles serves as an example. We define a simple model of a concentric plume with maximum concentration in the center and decreasing concentration to the outside in the style of a Gaussian air pollutant transport model (compare Giostra 1994), which is commonly used in local dispersion modelling. As our reasoning approach requires a qualitative representation, we define a boundary of the plume as some threshold concentration, which could be the minimum measurable concentration. The rate of concentration decreases with distance from the center and the radius of the plume are isotropic. Over time diffusion causes the plume to expand in space while its maximum concentration is decreasing. The plume travels in space and changes its travelling direction and speed only gradually. The gas concentration *c* in the plume at time *t* is modeled as a function of distance δ from the plume center

$$c_t(\delta) = Q_t \cdot e^{-\frac{dist^2}{\sigma_t}} \tag{11}$$

where Q_t is the maximum concentration in the center of the plume and σ_t describes the rate of concentration decrease towards the outside. Q_t can be modelled as

$$Q_t = f(t) = e^{x_0 + x_1 t + x_2 t^2} \tag{12}$$

with $Q_0 = f(0)$ representing the concentration immediately after the emission and $\forall t_0 > 0, \forall t_1 > t_0 : Q_{t_0} > Q_{t_1}$. σ_t can be modelled as

$$\sigma_t = z_0 + z_1 t + z_2 t^2 \tag{13}$$

with $\forall t_0 > 0, \forall t_1 > t_0 : \sigma_{t_0} < \sigma_{t_1}$. Coefficients z_0, z_1, and z_2 vary with atmospheric stability, i.e. unstable atmospheric conditions make the plume expand more quickly. The Gaussian plume travels with a certain speed into a certain direction.

3.2 Reasoning about a moving plume

Let us now assume a sensor that is able to measure gas concentrations above a certain threshold and passes through the plume moving on a straight line at constant speed. The size of the plume along the line upon which it is passed by the sensor depends on its proximity to the plume center. If the sensor passes the plume close to the fringe the length is essentially shorter than if it passes the plume through its center. Figure 4 illustrates how the sensor signal differs in length and amplitude depending on how close to the center of the plume the sensor passes through the plume.

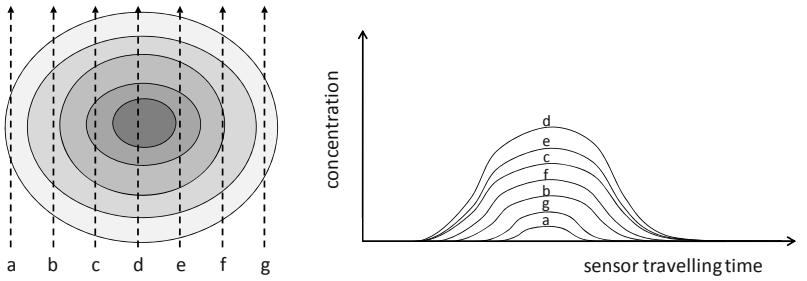


Fig. 4. The sensor signal differs in length and amplitude depending on how close to the center of the plume the sensor is passing through the plume.

The first and the last above-threshold measurements mark the time points of entering t_e and leaving t_l the plume. At some time between t_e and t_l, the measured value reaches a maximum *cmax*. We make the simplifying assumption that this maximum measurement is obtained in

$t_{c\max} = t_e + \frac{t_l - t_e}{2}$, although it may happen that the amount of above-threshold values after the maximum measured value is higher than of those before, if the plume expands quickly during the measurements. By making this assumption, we ensure that $cmax \leq Q_{t_{c\max}}$ and we can take

$$c_{t_{c\max}}(\delta) = Q_{t_{c\max}} \cdot e^{-\frac{dist^2}{\sigma_{t_{c\max}}}} \tag{14}$$

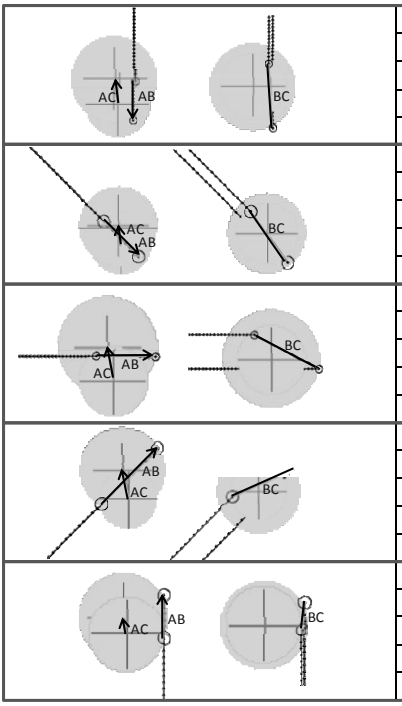
as a reasonable plume model for reasoning about distance between the center of the plume and the line along which the plume was passed by the sensor. This distance together with the radius of the plume at $t_{c\max}$ allows inferring the diameter of the plume along the line it was passed through by the sensor at $t_{c\max}$. The diameter roughly corresponds to the size of the plume along the line upon which it was passed by sensor *BC*, if we consider the size of the plume at $t_{c\max}$ as the mean plume size during the interval $[t_e, t_l]$. The qualitative distance value of *BC* can be used by the inference rules depicted in Figure 3.

To test our method in the described use case, we simulated Gaussian plumes using the Polyphemus Air Quality Modelling System (Mallet, et al., 2007) and used this data to simulate sensor measurements. We illustrate the qualitative spatio-temporal reasoning method by giving simulation results of a plume travelling to the North-North-West being approached by sensors moving into the directions south, south-east, east, north-east, and north. Figure 5 shows the plumes at t_e and t_l of these simulations overlaid with geographic space as the reference frame (left picture) and with the center of the plume as the spatial reference (right picture). The distances *AB*, *AC,* and *BC* are highlighted. For each simulation, *AC* was inferred from *BC* and *α*, *BC* was inferred from *AC* and *α*, and *α* was inferred from *AC* and *BC* using the inference rules depicted Figure 3. The true quantitative and qualitative distances and angles of the simulations and the inferred qualitative values are depicted in Figure 5. We will have a closer look at three reasoning examples.

Consider the following situation depicted in the first row of Figure 5: We have the reference distance *AB=19.176* that was travelled by the sensor. Let us assume we know from the Gaussian model and the sensor measurements that the observed plume is *Greater* in the diameter travelled by the sensor than *AB* (*BC=35.235*) and that the movement speed of the plume is *SlightlyLess* than the speed of the sensor. Then we can reason in

our *ΔABC* that the angle *α* must be *SlightlyObtuse* to *Obtuse*, i.e. the plume is moving into the opposite direction. Note that even if we were only able to say that the speed of the plume was *Less* to *Equal* the reasoning method would provide the same set of possible qualitative angles and thus no less information.

Looking at the second row in Figure 5, let us assume we know from the Gaussian model and the sensor measurements that the diameter of the plume travelled by the sensor is *SlightlyGreater* (*BC=37.348*) than the reference distance *AB=25.568*. Furthermore we roughly know based on presence of the plume at the current sensor location and the location of the emission source that the angle between sensor and plume movement *α* is *Obtuse*. Then we can reason about the speed of the plume and infer that the distance travelled by the plume during the observation is *Less* than *AB*.



	true quantitative	true qualitative	inferred qualitative
AB	19.176	eq	-
AC	12.784	sl	l, sl, eq, sg, g
BC	35.235	g	g
α	185°	o	so, o

	true quantitative	true qualitative	inferred qualitative
AB	25.568	eq	-
AC	10.227	l	l
BC	37.348	sg	sg, g
α	140°	o	sa, r, so, o

	true quantitative	true qualitative	inferred qualitative
AB	30.681	eq	-
AC	20.454	l	l, sl, eq, sg
BC	41.100	sg	sg
α	95°	so	sa, r, so, o

	true quantitative	true qualitative	inferred qualitative
AB	41.548	eq	-
AC	16.619	l	l, sl, eq, sg, g
BC	36.953	sl	l, sl
α	50°	a	a, sa

	true quantitative	true qualitative	inferred qualitative
AB	19.176	eq	-
AC	7.670	l	l, sl, eq, sg, g
BC	13.216	sl	l, sl
α	5°	a	a, sa

Fig. 5. Left: Plumes at t_e and t_l of simulations of sensor measurements of gas plumes (as described in section 2.1) overlaid using geographic space as the spatial reference (left picture) and using the center of the plume as the spatial reference (right picture). The lines correspond to the distances *AB*, *AC,* and *BC* in the triangle described in Section 2.1. Right: True qualitative and quantitative distances and angles in the triangle and the corresponding inferred qualitative values using the inference rules of qualitative trigonometry.

In the third row, the reference distance travelled by the sensor is *AB=30.681*. Let us now assume we know from meteorological data the speed and the direction of the plume movement. So we can calculate both the distance travelled by the plume *AC=20.454* (*Less*) and the angle between sensor and plume movement $\alpha = 95°$ (*SlightlyObtuse*). Now we can reason about the size of the plume. The diameter of the plume travelled by the sensor *BC=41.1* is *SlightlyGreater*.

These examples illustrate the advantages of qualitative trigonometry over quantitative trigonometry. In the first example, for instance, the knowledge about the speed of the plume is imprecise. However, even if we only know for sure that the sensor moves faster than the plume, the qualitative method yields useful results. The same applies to the second example, where the knowledge about α is imprecise. The third example suggests the use of quantitative trigonometry. However, in any case, a reasoning result based on quantitative trigonometry would suggest some precision which cannot be achieved, because the quantitative distances of the sides and angles will hardly be accurate as distance measurements and models inherit some imprecision and uncertainty.

4 Discussion

For planning a sensor route that captures a moving object in space and time it is necessary to approximately know when the plume will be where. Based on imprecise or incomplete information about the characteristics of the object, we present a method for inferring whether from the sensor perspective the plume was passing from behind, from the side, or from the front and whether the object is moving slower or faster than the sensor. The approach can be used in both the two- and the three-dimensional space. It provides the information required to adapt the sensor speed and the sensor direction to track the object and to successfully monitor the object. Future research is needed to develop an adaptive movement strategy for mobile sensors that utilizes the presented method to accomplish this task in the two- and in the three-dimensional space. Such a strategy should also take into consideration the following ideas resulting from this work:

- By varying the sensor speed and thus the reference distance *d* in multiple observations, it is possible to increase the precision of an inferred distance or angle.
- The grid representation of the inference rules depicted in Figure 3 reveals that there is less ambiguity in reasoning about the size of the object if the angle between sensor and object movement is *SlightlyAcute* to

Obtuse or the distance travelled by the object is *SlightlyLess* than the distance travelled by the sensor.

- Reasoning about the distance travelled by the object is only possible, if the size of the object along the line passed by the sensor is *Slightly-Greater* to *Greater* than the distance travelled by the sensor or the angle between sensor and object movement is *Acute* to *SlightlyAcute* (compare Figure 3).
- There is less ambiguity in reasoning about the angle between sensor and object movement if the size of the object along the line passed by the sensor is *Less* to *Equal* compared to the distance travelled by the sensor (compare Figure 3).

The presented method performs the reasoning based on the data measured by a sensor passing through the object only once. An ambiguous reasoning result can be refined by a reasoning based on the measurements of a sensor passing through the object again with modified speed or angle. A problem for future research is how to handle temporal variations of the object movement direction or speed over multiple reasoning steps.

Qualitative reasoning is particularly useful in the use case of monitoring a travelling gas plume, as the information about the movement of the plume to be used for reasoning is often imprecise. Data on wind velocity and wind direction, for example, may also come in qualitative categories, e.g. Beaufort 1, NE. The semantic integration of these qualitative categories and those involved in our method constitutes a research question for future work.

Another open issue for future work is the integration of more complex models of the observed object into the method. If we stay in the domain of atmospheric dispersion, the next step would be an anisotropic Gaussian plume and then more complex, e.g. concave shapes of plumes. These models could, for example, be extracted as oriented geometries out of numerical simulation fields. Estimating the size of the object along the line travelled by the sensor as input for the reasoning method will be more complex in these cases and the estimates are likely to be less precise than in the Gaussian model described, i.e. ranges of values.

5 Conclusion

This paper presented a method for qualitative spatio-temporal reasoning about moving objects from mobile sensor data based on qualitative trigonometry. We analysed the geometric configuration of the sensor and the observed object over time and came up with a simple trigonometric rela-

tionship between sensor movement, object movement, and object size that allows us to make inferences. We chose qualitative representations of distances and angles and qualitative reasoning using qualitative trigonometry in order to 1) account for the imprecision and incompleteness of the information used for inference and 2) provide the approximate reasoning results required by an adaptive sensor movement strategy for monitoring moving objects. We discussed the use case of monitoring a travelling gas plume using a mobile sensor. Finally, we came up with some interesting findings for consideration in the development of an adaptive sensor movement strategy and described some directions for future research.

Acknowledgements

This research is funded by the International Research Training Group on Semantic Integration of Geospatial Information of the DFG (German Research Foundation) GRK 1498. Edzer Pebesma contributed helpful ideas and discussion. Three anonymous reviewers provided valuable comments.

References

Cohn, A.G. and Renz, J. (2008) Qualitative spatial representation and reasoning. Foundations of Artificial Intelligence, 3:551-596.

Bian, L. (2007) Object-Oriented Representation of Environmental Phenomena: Is Everything Best Represented as an Object? Annals of the Association of American Geographers, 97(2):267-281.

Duckham, M., Nittel S., and Worboys M. (2005). Monitoring dynamic spatial fields using responsive geosensor networks. In Proceedings of the 13th annual ACM international workshop on Geographic information systems, pages 51-60. ACM.

Duckham, M., Stell, J., Vasardani, M., and Worboys, M. (2010) Qualitative change to 3-valued regions. In Sara Fabrikant, Tumasch Reichenbacher, Marc van Kreveld, and Christoph Schlieder, editors, Geographic Information Science, volume 6292 of Lecture Notes in Computer Science, pages 249-263. Springer Berlin / Heidelberg.

Dutta, S. (1990) Qualitative spatial reasoning: A semi-quantitative approach using fuzzy logic. In Design and Implementation of Large Spatial Databases, volume 409 of Lecture Notes in Computer Science, pages 345-364. Springer Berlin / Heidelberg.

Frank, A.U. (1996) Qualitative spatial reasoning: Cardinal directions as an example. International Journal of Geographical Information Science, 10(3):269-290.

Freksa, C. (1992) Using orientation information for qualitative spatial reasoning. In Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, volume 639 of Lecture Notes in Computer Science, pages 162-178. Springer Berlin / Heidelberg.

Giostra, U. (1994) Dynamical models of pollutant transport in the atmosphere. Aerobiologia, 10(1):53-57, Springer Netherlands.

Jiang, J. and Worboys, M. (2008) Detecting basic topological changes in sensor networks by local aggregation. In Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, pages 1-10. ACM.

Liu, J. (1998) A method of spatial reasoning based on qualitative trigonometry. Artificial Intelligence, 98(1-2):137-168.

Mallet, V., Quélo, D., Sportisse, B., Ahmed de Biasi, M., Debry, É., Korsakissok, I., Wu, L., Roustan, Y., Sartelet, K., Tombette, M., et al. (2007) Technical Note: The air quality modeling system Polyphemus. Atmospheric Chemistry and Physics Discussions, 7(3):6459-6486.

Patan, M. and Uciski, D. (2004) Robust activation strategy of scanning senors via sequential design in parameter estimation of distributed systems. In Parallel Processing and Applied Mathematics, volume 3019 of Lecture Notes in Computer Science, pages 770-778. Springer Berlin / Heidelberg.

Shi, M. and Winter, S. (2010) Detecting change in snapshot sequences. In Sara Fabrikant, Tumasch Reichenbacher, Marc van Kreveld, and Christoph Schlieder, editors, Geographic Information Science, volume 6292 of Lecture Notes in Computer Science, pages 219-233. Springer Berlin / Heidelberg.

Song, Z., Chen, Y.Q., Liang, J.S., and Ucinski, D. (2007) Optimal mobile sensor motion planning under non-holonomic constraints for parameter estimation of distributed systems. International Journal of Intelligent Systems Technologies and Applications, 3(3):277-295.

Walkowski, A. C. (2008) Model based optimization of mobile geosensor networks. In The European Information Society, Lecture Notes in Geoinformation and Cartography, pages 51-66. Springer Berlin / Heidelberg, 2008.

Worboys, M. and Duckham, M. (2006) Monitoring qualitative spatiotemporal change for geosensor networks. International Journal of Geographical Information Science, 20(10):1087-1108.

Connecting R to the Sensor Web

Daniel Nüst, Christoph Stasch, Edzer Pebesma

Institute for Geoinformatics, University of Muenster
Muenster, Germany
{daniel.nuest, staschc, edzer.pebesma}@uni-muenster.de

Abstract. Interoperable data exchange and reproducibility are increasingly important for modern scientific research. This paper shows how three open source projects work together to realize this: (i) the R project, providing the *lingua franca* for statistical analysis, (ii) the Open Geospatial Consortium's Sensor Observation Service (SOS), a standardized data warehouse service for storing and retrieving sensor measurements, and (iii) sos4R, a new project that connects the former two. We show how sos4R can bridge the gap between two communities in science: spatial statistical analysis and visualization on one side and the Sensor Web community on the other. sos4R enables R users to integrate (near real-time) sensor observations directly into R. Finally, we evaluate the functionality of sos4R. The software encapsulates the service's complexity with typical R function calls in a common analysis workflow, but still gives users full flexibility to handle interoperability issues. We conclude that it is able to close the gap between R and the sensor web.

1 Introduction

As the whole process of environmental analysis is moving to the internet (creating a model web (Geller 2008)) the data sources are also, which results in a growing amount of spatially distributed sensor data publicly available through standardized web service interfaces. Yet, client applications to access, to analyze, and to visualize this data lag behind compared to spatial data in a raster or vector format. This is partly due to a gap between the research fields of sensor web and data analysis or geostatistics.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_12,

The former group devotes itself to the development and implementation of standards for sensor data storage and exchange in the realm of Open Geospatial Consortium (OGC) or International Organization for Standardization (ISO) – a rather technical undertaking often based in computer science. The latter group, often domain specialists, actually analyzes the data to infer about the processes that generated it. This organizational and personal difference (in the sense that few people work in both research areas) results in a separate development of tools and knowledge about service-based data retrieval and data processing. We try to narrow this gap by providing a tool that can benefit both communities and further facilitate interdisciplinary collaboration and research. A motivation for this work is to increase acceptance of the Sensor Web ideas and tools and allow more data analysts and modellers to benefit from the Sensor Web.

The general driving force is to motivate analysts and researchers to adopt open source software and open standardized interfaces and to conduct reproducible research. To achieve this goal, the main contribution of this work is a Sensor Observation Service (SOS) (Na et al. 2007) client written in R (R Development Core Team 2010) that addresses the translation of the required observation models to R data structures. We investigate if a (partial) gap closure can be done by a single piece of software as the decisive chain link in a sequence of existing tools.

The term reproducible research was first proposed by Jon Claerbout (Fomel and Claerbout 2009) to tackle the issues of current scientific publications in the domain of computational sciences, where the components necessary to recreate the presented results are completely provided. It "refers to the idea that the ultimate product of research is the paper along with access to the full computational environment used to produce the results in the paper such as the code, data, etc. necessary for reproduction of the results and building upon the research" (Reproducible Research Planet 2010). The test set-up is a critical component in experimental sciences. Although in computational science one can expect that the same code and data result in the same outcome, there is still the aspect of sharing both of these. By using tools that directly connect code and publication, reproducible research does not only allow others to replicate results, but also the original authors themselves. We want to support the analysis part of reproducibility from a technical standpoint.

The remainder of this paper is structured as follows. Sections 1.1 to 1.3 present the basic concepts and components upon which the product is built and related work. Section 2 introduces a short use case whose exemplary steps are used to develop the requirements presented in section 3. Afterwards we describe the concept of sos4R in sections 4 to 5 and conclude

with an evaluation (section 6). An agenda for future work is outlined in section 7.

1.1 The R project

R is an emerging language and environment for statistical computing and graphics (Vance 2009). It provides a wide range of statistical techniques and analysis functions as well as capabilities of graphical plotting that stand out and partly underlie its success. A complete description of R's functions and its user and developer communities (ranging from finance to biomedicine) is not possible here, but the most important features from the sos4R perspective are: R is freely available under an open source license; it is extensible via packages, which are small (to large) units of code; and data that can be comfortably added to an R environment.

The *Sweave* document format (Leisch 2005) supports the concept of weaving, where the text and the analysis code are in the same document (introduced by Knuth under the term "literate programming" (Knuth, 1984)) and is particularly well suited for reproducible research as a single document generates text and results. The package *cacher* implements cached computations as a framework for reproducible research (Peng 2008). By plugging in the most important data source of the OGC Sensor Web into R, a wide range of data becomes readily accessible to the large community of R users.

1.2 The Sensor Observation Service

Providing data via web services supersedes the need for local file copies, which might become outdated. Also, a flexible filtering of data on the service side reduces the communication necessary to download the data. The OGC SOS is a common specification for a web service providing pull-based access to sensor observations and sensor descriptions. It is part of the Sensor Web Enablement Initiative (SWE) (Botts et al. 2008). The observations can be provided in real-time or as archived data sets. They can be subsetted in a flexible way. For example, a query like "Provide observations from water gauge Cologne_Rhine_1 for the time period from 01/01/2010 to 31/08/2010 where the water level is above 5 meters" can be created in a common format and sent to the SOS which returns only matching observations.

As the SOS always provides the observations in a common format based on Extensible Markup Language (XML)[1], namely Observations & Measurements (O&M) (Cox 2007a), clients do not have to adapt different data formats when integrating observations. An observation consists of information about the geographic feature which is observed, the time when the observation was taken, the sensor, the observed phenomenon, and the observation's result. The SOS relies upon the Sensor Model Language (SensorML) (Botts 2007) for providing sensor metadata like the sensor's position, calibration information, and inputs and outputs of the sensing process. By connecting the powerful R environment with the SOS, a huge amount of features for analysis and visualization become readily accessible to the community of SOS users.

1.3 Related Work

The OGC offers widely used service standards apart from SOS, for example Web Features Service[2] (WFS), Web Coverage Service[3] (WCS) and Web Map Service[4] (WMS). A connection is possible in R through the *rgdal* (Keitt et al. 2010) package which provides bindings to the Geospatial Data Abstraction Library (GDAL)[5]. However, these do not (directly) support typical sensor data, like in situ measurements, but vector and raster ("image") data. WFS could provide sensor observations as point features, but that is not intended.

With this distinction, related work can be seen from different perspectives. First there are other packages in R which allow the comfortable download of near real-time data directly into R data structures. Second, there are other clients for Sensor Observation Services.

For the former, one can find a few packages, for example the *GEOquery* package of the Bioconductor project (Davis 2007) and *WDI* (Arel-Bundock 2010) package, which allow access to the Gene Expression Omnibus repository of gene expression experiments and access to the World Bank's World Development Indicators respectively. The *quantmod* package (Ryan 2008) provides functions for downloading stock exchange data. However, these use proprietary interfaces for a specific data base.

The latter perspective comprises mostly viewers of varying abilities:

[1] http://www.w3.org/XML/
[2] http://www.opengeospatial.org/standards/wfs
[3] http://www.opengeospatial.org/standards/wcs
[4] http://www.opengeospatial.org/standards/wms
[5] http://www.gdal.org/

- simple viewers, like the OpenOOIS.org Map Viewer[6] or the SOS client for OpenLayers[7], which show the positions of sensors on a map and can include links to retrieve observational data;
- advanced viewers, like 52°North Clients (52°North, 2010) or GeoCENS Sensor Web Browser (Liang et al. 2010), which include (three dimensional) visualization of sensors' positions on a map, tabular, or graphic display of data; and
- plug-ins to geographic information systems, like ArcGIS and uDig (52°North 2010) or gvSIG (Tamayo et al. 2009), which allow the import, subsequent processing with the respective software, and most importantly combining with a large number of other data sources.

2 Application Example

To illustrate the use of the sos4R package, we present the use case of a researcher modelling local weather data. She requires temperature measurements for the region and the time frame of interest. The workflow consists of the following steps:

1. Create a connection to a SOS
2. Set up query parameters
3. Request observation data
4. Analyse and visualize data with R functionalities (not part of this work).

For the sake of simplicity this example does not utilize the (arbitrary) spatial, temporal, and value-based querying features of the SOS.

Listing 1 and Figure 1 show the complete code and the result of the analysis, a temperature plot. The code and figures are also available from http://www.nordholmen.net/sos4r/agile2011.

Listing 1. Example analysis with sos4R (lines starting with "#" are comments)

```
library("sos4R"); library("xts")

# 1. step
weathersos = SOS("http://v-swe.uni-muenster.de:8080/WeatherSOS/sos")

# 2. step
station <- sosProcedures(weathersos)[[1]]
temperatureOffering <- sosOfferings(weathersos)[["ATMOSPHERIC_TEMPERATURE"]]
temperature <- sosObservedProperties(temperatureOffering)[1]
```

[6] http://www.openioos.org/real_time_data/gm_sos.html
[7] http://www.openlayers.org/dev/examples/sos.html

```
september <- sosCreateTimePeriod(sos = weathersos,
                 begin = as.POSIXct("2010-09-01 00:00"),
                 end = as.POSIXct("2010-09-30 00:00"))

# 3. step
obsSept <- getObservation(sos = weathersos, observedProperty = temperature,
                 procedure = station,
                 eventTime = sosCreateEventTimeList(september),
                 offering = temperatureOffering)
data <- sosResult(obsSept)

# 4. step
summary(data); data[1:2,]; names(data)
# create time series
tempSept <- xts(x = data[["urn:ogc:def:property:OGC::Temperature"]],
         order.by = data[["Time"]])
# calculate regression (polynomial fitting) and plot
temp <- data[["urn:ogc:def:property:OGC::Temperature"]]
time <- as.numeric(data[["Time"]])
x = loess(temp~time, na.omit(data), enp.target = 10)
plot(tempSept, main = "Temperature at Station One",
                 xlab = "Time", ylab = paste("Temperature in", attrib-
utes(temp)[["unit of measurement"]]),
                 major.ticks = "weeks")
lines(data$Time, x$fitted, col = 'red', lwd=3)
```

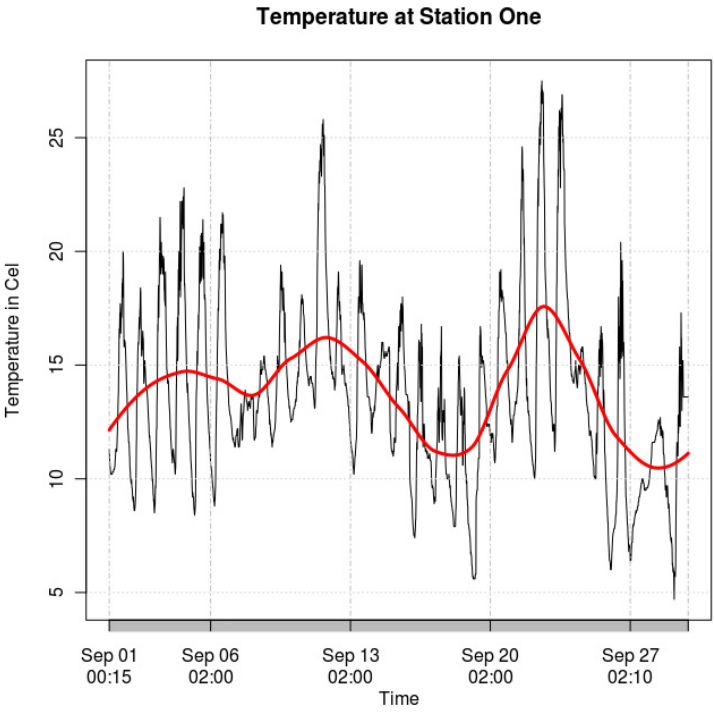


Fig. 1. Time series plot of exemplary analysis showing temperature values from September 2009 together with a regression line

3 Requirements Analysis

The sos4R client naturally implements the "core profile" from the SOS specification which comprises the three basic operations for information retrieval, namely *GetCapabilities* for retrieving service descriptions, *GetObservation* for querying observation data, and *DescribeSensor* for retrieving sensor metadata encoded as SensorML.

The software must allow for the comfortable creation and exploration of these operations including exceptional events. This should also be possible for non-experts in the sense that the user shall not be closely familiar with the whole range of OGC specifications related to SOS and to a certain extent without even knowing the specification of the core operations themselves. The request creation shall be supported by sensible default settings, but still allow full flexibility for advanced settings. The responses shall be transformed to common R data structures and thereby allow immediate subsequent analysis. Convenience functions shall be provided to explore, e.g. the capabilities of a service.

As the SOS standard has been made suitable for almost any possible type of observation, the possible encodings it may return are too large to cope with for a typical client. O&M as response format is very flexible and supports many, sometimes uncommon or rarely needed, possibilities. Thus, an XML Schema[8] profile which limits the processable markup is needed. In the absence of a generic simple observation profile, the O&M profile of the 52°North SOS[9] was chosen. This SOS is a stable, compliancy tested open source implementation and the profile suffices for common use cases. In brief, this profile restricts the different possibilities for encoding temporal, spatial, and result information to certain pre-defined types. Naturally, not all existing SOSs support the same encodings and filters as in the 52°North profile or support them completely themselves. That is why exchangeability must be seen as the most important requirement. The user shall be given flexible tools to exchange only the necessary processing steps to include her own data markup.

A potential problem was identified early during the development. If the user ought to be "shielded from" the SOS specification, in order to lower the threshold for new users, the difference in technical terms of a SWE expert became apparent. Naturally used terms like "procedure" or "observed property" were introduced for clarity in specifications, but are not the typical "laymen" terms of professionals from other domains.

[8] http://www.w3.org/XML/Schema
[9] https://wiki.52north.org/bin/view/Sensornet/SensorObservationService

Thus, a qualitative survey was conducted with a small number of expert users with either a lot experience with SOS or none at all, or vice versa, having high experience with R or close to none.

The survey stated general tasks, e.g. "Create (a connection to) a SOS instance, e.g. based on the URL [...]." or "Request observation data of a known location and phenomenon [...] for a certain time period". The participants were asked to outline a short R script of how they would expect a SOS client in R could be used. The questionnaire was accompanied by a short introductory description of R and SOS.

The following could be observed for experienced R/inexperienced SOS users: a typical user immediately wanted to plot and run calculations on the returned object, showing that she/he was not aware that the response contains more (meta-) information than just the plain data as a table for example. None of the subjects used the typical SWE terms, but rather commands like "read" to request data, "sensors" for procedures, and "phenomenon" for observed properties, even "location" referring to a feature of interest. Different results are naturally possible with people from other domains. Furthermore, it became also very apparent that the users wrote rather simple function calls with all the settings they require as arguments, i.e. no higher class objects as input parameters.

Experienced SOS users followed the workflow laid out by the SOS operations, starting with a GetCapabilities request and then starting a GetObservation operation with the respective parameters. As the threshold is considerably lower for these types of end users, we do not see any requirement for special accommodation.

An interpretation of the results allows the following statements with regard to the user interface. Functions to directly access the original SOS operations with the SWE terms must be provided for users who are familiar with these standards. An argument against other terminology is the confusion of users moving from simple use cases to more advanced levels or going deeper into Sensor Web concepts. Thus, more abstract functions containing a translation between domains, i.e. the SWE domain and the respective application domain, can be implemented based on further user testing or based on dictionaries like in Annex B to O&M (Cox 2007a). These simpler functions, named after common functions like “read” encapsulate the complexity of standards where full features are not needed.

4 Software Design

Three main challenges have been identified: designing a web service client in a command line based environment, ensuring exchangeability to enable the integration of different SOS profiles, and mapping the conceptual models of OGC specifications to R classes.

4.1 User Interface

The basic R user interface is a command line prompt. It allows help documents to be opened, provides command completion and a history, and it can be used in the same way on all platforms. Syntax highlighting for R source files is available for many standard editors, but there are dedicated programming environments as well. Nevertheless, the primary target user interface for sos4R is the console and all functions are designed with regard to that. This is a limiting factor to the requirement of exploration which profits from a graphical user interface. We suggest different tools, like sensor catalogues and registries (Jirka et al. 2009), for exploring a service's capabilities for the first time.

4.2 Component Exchangeability

Extensibility is a crucial advantage of open source software, but not all users might be able or willing to download the source code of a program to introduce minor changes to get software to work for their use case. For a client of a web service standard that does not (and cannot, see section 3) implement all possible features out of the box, users must be able to adapt the processing chain. Therefore, three key components in a SOS connection are easily exchangeable.

- Parsers: parsing functions create R objects based on textual encoding. These are used by the service interface receiving responses.
- Converters: conversion functions transform a given character string representation of a data value into the correct atomic data type in R, e.g. double.
- Encoders: encoding functions translate an R representation of an object to external objects, often character strings, so that they can be transferred to other software. These are used by the service interface sending requests.

Similar to a template method pattern (Freeman et al. 2004) in object oriented programming, the basic workflow is fixed but the single steps have interchangeable components. The sequence in a client server setup can be abstracted to five steps (see Figure 2):

1. building a request (based on user input);
2. transforming the request object to a transferable format – the encoding;
3. sending it to the service using a certain method and receiving the response – the transfer;
4. processing the response (includes converting) – the parsing; and
5. analyzing the data.

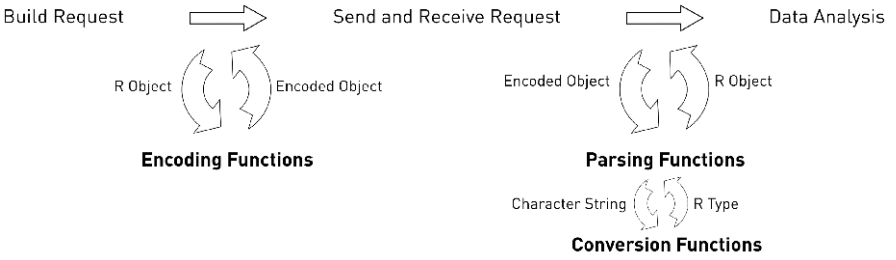


Fig. 2. The sensor data analysis workflow: steps and exchangeable components

The code mechanism takes advantage of the possibility to store function objects as variables. The functions performing the first and last step are provided in a named list when creating a SOS connection. This list can be exchanged by a user. The simple example in listing 2 illustrates this.

Listing 2. Exchange of parsing function for DescribeSensor operation.

```
myParseSensorML <- function(obj) {
        root <- xmlRoot(obj)
        return(xmlName(root))
}
mysos = SOS(url = "http://sos.org/sos",
        parsers = SosParsingFunctions("DescribeSensor" = myParseSensorML))
```

In this example, the user does not want the response object of the DescribeSensor operation to be stored completely, but only keep the name of the root element. Adding the name of this function to a named list, where the names correspond to an XML element name or the operation that is performed, makes parts of the parsing exchangeable. This list is created

with a utility function that combines default functions with the user-defined ones, which is useful if only a certain element must be handled specially.

The same is possible for the conversion functions where possible names are the definition of a data field or its unit of measurement.

We took a different approach for encoders, because the possible encodings and request documents are already defined in the service. SOS supports two HTTP[10] based bindings: GET with a key-value pair encoding in the URL[11]; and POST with XML encoded requests. Therefore a single generic method suffices for each binding and is added to the encoders list named after the connection method. This method must then be implemented for all objects which need to be encoded and R resolves the correct method based on the input objects.

4.3 Mapping OGC Specifications to R Classes

The transfer of OGC requests and data structures to R objects is a key aspect. We describe a selection of elements and operations from the various specifications in this section. Some general considerations that originate partly in the manual mapping procedure are as follows.

- The 52°North SOS profile has been used as a guideline for the included features. Many optional (but rarely used) elements and attributes of specifications are not included.
- XML elements are mapped to S4 classes in R, which in turn contain slots with contained elements (potentially as lists) and attributes.
- The extra layer of element types is omitted for brevity.
- Extensions of the XML type system, e.g. substitution groups or abstract classes, are only partially implemented.

The implementation of specifications consists of four components: classes (in an object oriented sense) for XML elements, functions for parsing/decoding (XML) character representations, functions for encoding classes (to XML), and functions for creating instances of the classes. The last functions preferably utilize only R data types and structures. Differences in encodings, as some elements are only possible in certain encoding types, are not explicitly mentioned.

[10] http://www.w3.org/Protocols/

[11] It must be noted that the GET binding is not part of the official standard, as that section was accidentally left out, but defined in a best-practice paper available at http://www.oostethys.org/best-practices/best-practices-get.

4.3.1 OGC Web Services Common

The OGC Web Services Common specification (OWS Common) specifies "[...] many of the aspects that are, or should be, common to all or multiple OWS interface Implementation Specifications" (Whiteside 2007). OWS Common forms the basis for all OGC service specification including the SOS. It comprises operations (request and response) and their transfer (using HTTP), encodings, parameters, and data structures. This specification is implemented generically and the classes and methods could be extracted to a separate package and reused by other packages. Some elements not used in the SOS specification are omitted.

GetCapabilities is the core operation of OWS Common and both the request and response are implemented completely. *ExceptionReports*, which wrap exceptions to handle erroneous states in the service or illegal requests, are fully covered (see Figure 3).

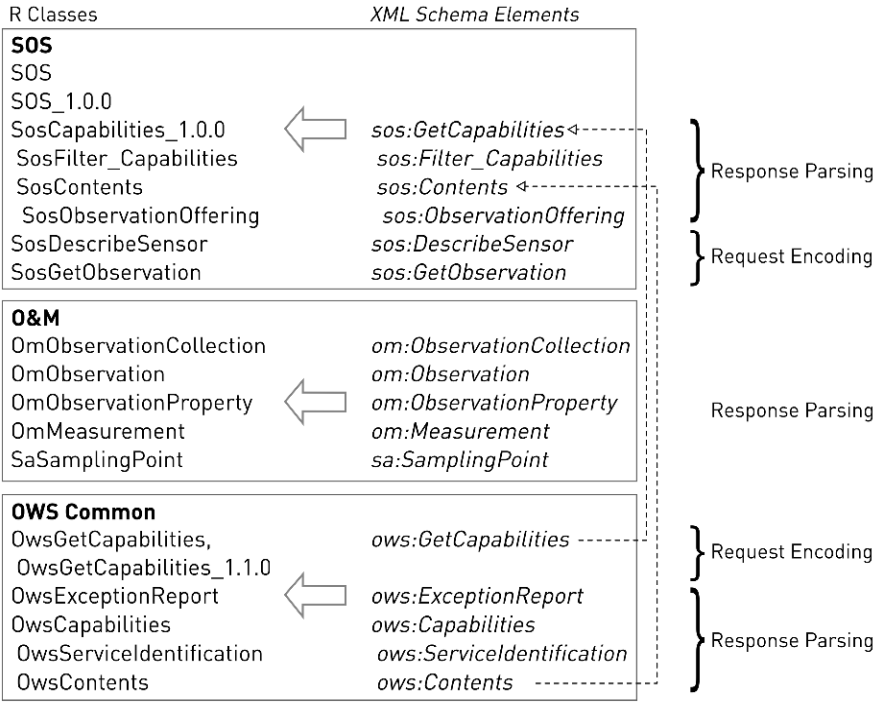


Fig. 3. Overview of important mappings from XML elements to R classes

4.3.2 Sensor Observation Service

Classes are implemented equivalent to the important objects in connecting and retrieving data from a SOS (see Figure 3). These are the class SOS itself and classes for the core operations.

The virtual[12] class *SOS* represents the concept of a connection to a SOS and can be used with different functions for extracting information from a SOS. It contains slots for elements that are unlikely to change between versions. It has a sub-class, *SOS_1.0.0*, with additional or specifically redefined slots for all required information for a connection to a SOS instance implementing version 1.0.0 of the specification. This structure loosely follows the adapter pattern (Freeman et al. 2004), where new service interfaces can be supported by adding a new adapter class, but common methods are reused and the user experience does not change. However, the distinction between versions also allows easy adaption of future specifications.

The classes for core operation requests contain all request parameters in a suitable format, i.e. (lists of) objects of classes presented here.

4.3.3 Geography Markup Language

The Geography Markup Language (GML) (Portele 2003) provides basic spatial and temporal data types in the service requests and responses of a SOS. Only the elements and attributes that we see as most common are included, e.g. not the GML metadata element. At some points flattening is used to remove levels in the class model by not implementing an extra class for a type or element. For example, the *description* element is not a class, but just a character slot to hold the content of a *description* element. Otherwise, the full element hierarchy is transferred into R classes. Abstract schema elements are virtual classes.

Basic spatial classes are implemented, e.g. *Point*. The structure for more complex elements like *LineString* or *Polygon* is laid out in the code. Temporal elements are represented with classes for time instants and time periods, which eventually hold an object of class POSIXt, the R class for calendar dates and times.

4.3.4 Observation and Measurements, and Sampling Features

The specification Observations and Measurements consists of two parts: part one defines an observation schema (Cox 2007a); part two defines sampling features (Cox 2007b). Figure 3 shows the classes for observation

[12] The R term for the notion of abstract classes.

collection, observations, and measurements. The specifications variety is resembled in the slot types of classes, which might accept any object type. Where possible, fixed structures are mapped directly, e.g. the *observedProperty* of an observation is an object of class *SwePhenomenonProperty*. *SamplingPoint* is the only mapped element of part two of the specification. It is commonly used to encode positions of sensors.

The goal of the workflow is to make the actual values of an observation response directly usable in R. Therefore, the *result* element of each *Observation* is parsed directly into an R *data.frame* object (see section 4.3.5).

4.3.5 SWE Common

The SWE Common specification is part of SensorML (Botts, 2007) and provides common data types used in the several sensor related specifications. The amount of classes from this specification is rigorously limited to essential ones, because SWE Common uses many abstract type declarations and possibly very complex nesting. The *CompositePhenomenon* holding a list of *Phenomenon* elements is supported. The classes respectively elements *Phenomenon* and *TextBlock* are essential for the parsing of a *Values* element, which itself is not a class but parsed directly to a *data.frame*. The former two elements contain metadata (amongst others, units of measurement and markup information). The latter contains the actual data values. A *data.frame* is a matrix-like structure with indexed columns of different types (like numeric, character, or list) and indexed rows. All metadata attributes are attached to the *data.frame* object.

4.3.6 OpenGIS Filter Encoding

OpenGIS Filter Encoding (Vretanos 2005) specifies a general query language in XML. SOS implementations can indicate the supported filters in the service metadata description in the element *Filter_Capabilities*. It is completely implemented so that users can browse an instance's supported filters. Functions help in the comfortable creation of the most common filters in a GetObservation operation. These are temporal filtering in the *eventTime* element and spatial filtering with a bounding box in the *featureOfInterest* element.

Temporal filtering operands are after, before, during, and equals. The operand classes contain a slot for the respective sensible time element of GML, for example a *TimeInstant* for a “before” query.

Spatial filtering operands are contains, intersects, and overlaps. The operand classes contain a slot for spatial elements of GML.

Result filtering based on parameter values is only possible via manual creation of the property filters.

5 Implementation

The presented concepts and requirements are implemented in the project sos4R[13], which is published as open source software under GNU General Public License[14] (GPL).

The software relies on two packages that both originate from the Omegahat Project for Statistical Computing (Temple Lang 2000). They are available under BSD license[15]. The first is package *XML* (Temple Lang, 2010) and it provides functions for parsing XML, both document-based (which is used) and event-driven approaches, and creating XML. The second is package *RCurl* (Temple Lang 2007), a package for composing HTTP request, i.e. GET and POST operations, to web servers. It builds upon the powerful library *libcurl*[16] and its extensive list of features, like HTTPS, cookies, and authentication, which can be exploited by expert users when creating a SOS connection. A test of the package *XMLSchema*[17], which allows automatic creation of R classes and conversion functions from XML objects to objects of these classes yielded technical problems, as the package cannot handle the large number of (partly circular) references in the schemata.

The implementation makes extensive use of S4 classes (Genolini 2008). These allow object-oriented programming in R, including type safety, inheritance, and encapsulation (also of the construction of objects).

Default values of parameters are based partly on personal experience, the aforementioned survey (e.g. the temporal operator), and the metadata description of service and contents. The number of required parameters could thereby be lowered to one in the case of the GetObservation operation, i.e. the offering.

Convenience functions, which can be used to create the most common elements solely in R code, serve the requirement of encapsulating the large-scale OGC specifications. For instance, there is a function for time periods, which accepts R classes for time (POSIXt) for begin and end times.

[13] http://www.nordholmen.net/sos4R/
[14] http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
[15] http://www.opensource.org/licenses/bsd-license.php
[16] http://curl.haxx.se/
[17] http://www.omegahat.org/XMLSchema/

The software was successfully tested with Sweave and allows the original response documents to be saved for archiving and reproducibility.

6 Discussion and Conclusion

In this paper, we presented the concept and implementation of the connection of the statistical analysis environment R to the Sensor Web's data provision service, the SOS. It supports the user in creating requests for observational data based on the service description metadata. The requests include flexible subsetting (thematic, spatial and temporal) for volume reduction of transferred files. For common use cases, this is possible using only R calls and without any contact to the actual request mechanism or documents that were sent and received. Thanks to the online data storage and open data format, users are not restricted to specific file formats, integration of most up-to-date data (potentially directly from the source holder) is easy and observations can even be requested in near real-time. The existing R tools for reproducible research are complemented well by sos4R and reproducibility can be increased.

The exchangeability features presented in section 4.2 are part of a more abstract common methodology for statistical analysis of sensor observations. The use case in section 2 and Figure 2 contain a common workflow for the import and processing of data from the Sensor Web into an analysis environment. There are fixed, ordered steps due to dependencies of the operations but also flexible components if necessary.

The component exchangeability is powerful and supplies the required degree of flexibility. The mapping of SOS and O&M specifications, including the data structures from SWE Common, works well for all tested use cases. The presented design decisions are valid. However, a complete modelling of SWE Common elements like multidimensional complex data arrays, to R classes, could allow more flexibility. This could comprise coercion functions to lists or time series classes. The automation of that coercion is cumbersome (for example automatic detection of the attribute that holds temporal information) and not exploited yet.

A similar case is filtering, where a full R implementation of (property-based) result filtering can assist users that are not familiar with XML and details of specifications. Here we see an especially high gain with an integration of the spatial, temporal, and upcoming spatio-temporal classes from other R packages as filtering input.

Shortcomings of the software can be found in few areas. The exploration capabilities based on service metadata descriptions are not visual and require previous knowledge of available information.

Generic standards limit the interoperability. We try to compensate with swappable segments in the analysis workflow until future versions of the standards supply improvement. A considerable deficit is the integration into spatial, temporal, and spatio-temporal data structures, which is merely indirect. An automatic integration of the downloaded data structures into suitable spatial or temporal data structures would be optimal, for example using the packages *sp* (Bivand et al. 2009) or *xts* (Ryan 2010). This requires common markup of data and an internal logic which can automatically detect temporal, rational, or ordinal variables. Beyond that, coherent spatio-temporal data structures are not supported. We trace this back to a general lack of integrated spatio-temporal data structures and analysis tools.

The client was successfully tested with several SOS instances based on implementations by 52°North[18], OOTethys[19], and Degree[20]. Other (open and closed source) implementations exist and must be tested for too, as soon as public test instances are available (for example Mapserver[21], istSOS[22]). The limitation of many of the specifications to the 52°North SOS profile did not prevent connection to other services. In fact, only expected adjustments for unknown data fields (converters) had to be made. Several of the R classes implemented in this work, in particular those for OWS Common (GetCapabilities) and specifications related to GML can be reused when an R client for another OGC web service is needed.

The software performed well during development, but further performance testing is required, especially with large data sets. Regarding the processing of response documents in R, the available memory is a limiting factor which could be handled by event-based parsing techniques. Regarding the data transfer, O&M might not be the appropriate format for massive data sets. Instead, a SOS could respond in a binary compressed format, like netCDF[23] for gridded data, and the user adapts the parsing function to the respective import method of that format. Our pragmatic approach reveals a lot of challenges and pitfalls of current software systems. The Sensor Web community focuses on developing standards and services.

[18] https://52north.org/communities/sensorweb/sos/
[19] http://www.oostethys.org/
[20] http://wiki.deegree.org/deegreeWiki/deegree3/SensorObservationService
[21] http://mapserver.org/ogc/sos_server.html
[22] http://istgeo.ist.supsi.ch/software/istsos/
[23] http://www.unidata.ucar.edu/software/netcdf

In our experience there are content providers as early adopters, but not many actual analyses are based on SOS data. We see the practical approach to provide a tool to other researchers to support collaboration and reproducibility as sound and viable. sos4R is the first SOS client for a software environment that focuses on coherent analysis of the data (but still includes visual presentation) and certainly is a novelty in the field.

7 Outlook and Future Work

Implementing the SOS core profile was the first step taken. However, we also see potential in implementing the transactional profile for which use cases spanning different scientific areas shall be developed. Output of analysis, e.g. some spatial, temporal, or spatio-temporal interpolation or forecast done in R could provide a feeding layer from a variety of specific sources. Or (intermediate) results of an analysis could be published and archived in a SOS together with R scripts as procedures.

The list of features for enhancement is lengthy, so only a few ideas shall be listed here: the plotting of observation offerings or service capabilities on a map; a stronger connection with package *sp* (e.g. coercion functions between GML spatial classes and sp classes); and a better connection with time series classes in R (e.g. direct conversion of a downloaded time series).

A related research topic is the modelling of spatio-temporal data in R: at the time of writing, R support for handling of spatio-temporal data in a fully referenced way is in its infancy. Another topic of a more general scope is the development of a simple observation profile for O&M. A simple profile can limit the possible data types and structures for the sake of easier interoperability.

The sos4R package was tested with available open source SOS implementations. To aid wide adaptation, upcoming service implementations must be continuously tested. If new service instances contain new result markups, their processing can be added to the code by any user thanks to the open source strategy. The next step regarding the package is to discover how the software is adapted by users. A long term goal of an open source project naturally is to build up a solid user and developer base. In the best case there is a large overlap between these two groups, but the success also depends on the adaptation by data providers and the number of SOS with interesting data. That is why an overall objective is to motivate content owners and analysts to make their data available through the SOS interface.

Acknowledgements

This work was generously supported by the 52°North Student Innovation Prize for Geoinformatics 2010. We also thank the reviewers for their valuable comments.

References

52°North - Initiative for Geospatial Open Source Software (2010) https://-52north.org/, Last date accessed 2010-10-20.

Arel-Bundock, V. (2010) WDI: Search and download data from the World Bank's World Development Indicators', http://cran.r-project.org/package=WDI.

Bivand, R. S. Pebesma, E. J. (2008), Applied spatial data analysis with R, Springer, NY, http://www.asdar-book.org/.

Botts, M. (2007) OGC Implementation Specification 07-000: OpenGIS Sensor Model Language (SensorML), Technical Report, Open Geospatial Consortium.

Botts, M., Percivall, G., Reed, C. and Davidson, J. (2008) OGC Sensor Web Enablement: Overview and High Level Architecture, in S. Nittel, A. Labrinidis and A. Stefanidis (Eds.), GeoSensor Networks, Vol. 4540 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 175–190. http://dx.doi.org/10.1007/978-3-540-79996-2_10.

Cox, S. (2007*a*) OGC Implementation Specification 07-022r1: Observations and Measurements - Part 1 - Observation schema, Technical Report, Open Geospatial Consortium.

Cox, S. (2007*b*) OGC Implementation Specification 07-022r3: Observations and Measurements - Part 2 - Sampling Features, Technical Report, Open Geospatial Consortium.

Davis, S. and Meltzer, P. S. (2007) GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor, Bioinformatics 23(14), pp. 1846–1847, http://dx.doi.org/10.1093/bioinformatics/btm254.

Fomel, S. and Claerbout, J. F. (2009) Guest editors' introduction: Reproducible Research, Computing in Science and Engineering 11, pp. 5–7.

Freeman, E., Freeman, E., Bates, B. and Sierra, K. (2004) Head First Design Patterns, O'Reilly Media, http://www.worldcat.org/isbn/0596007124.

Geller, Gary N., M. F. (2008) Looking Forward: Applying an Ecological Model Web to assess impacts of climate change, Biodiversity 9(3&4).

Genolini, C. (2008), A (Not So) Short Introduction to S4', http://cran.r-project.org/other-docs.html.

Jirka, S., Bröring, A. and Stasch, C. (2009), Discovery Mechanisms for the Sensor Web, Sensors 9, 2661–2681, http://www.mdpi.com/1424-8220/9/4/2661/.

Keitt, T. H., Bivand, R., Pebesma, E. and Rowlingson, B. (2010) rgdal: Bindings for the Geospatial Data Abstraction Library, http://CRAN.R-project.org/package=rgdal

Knuth, D. E. (1984) Literate Programming, The Computer Journal 27, pp. 97–111.

Leisch, F. (2005) Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis, http://www.stat.uni-muenchen.de/~leisch/Sweave/.

Liang, S., Chang, D., Badger, J., Rezel, R., Chen, S., Huang, C. and Li, R. (2010) GeoCENS: Geospatial Cyberinfrastructure for Environmental Sensing, Extended Abstracts for Presentation at GIScience 2010. http://www.giscience2010.org/pdfs/paper_219.pdf.

Na, A., Priest, M., Niedzwiadek, H. and Davidson, J. (2007) OGC Implementation Specification 06-009r6: Sensor Observation Service, Technical Report, Open Geospatial Consortium.

Peng, R. D. (2008), Caching and Distributing Statistical Analyses in R, Journal of Statistical Software 26(7), http://www.jstatsoft.org/v26/i07/.

Portele, C. (2003), OpenGIS Geography Markup Language (GML) Encoding Standard 07-036, Open Geospatial Consortium.

R Development Core Team (2010), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org.

Reproducible Research Planet (2010). http://www.rrplanet.com/, Last date accessed 2011-01-03.

Ryan, J. A. (2008), Quantitative Financial Modelling & Trading Framework for R, http://www.quantmod.com, Last date accessed 2011-01-01.

Ryan, J. A., Ulrich, J. M. (2010) xts: Extensible Time Series, http://cran.r-project.org/package=xts.

Tamayo, A., Huerta, J., Granell, C., Diaz, L. and Quiros, R. (2009), 'gvSOS: A New Client for the OGC Sensor Observation Service Interface Standard, *Transactions in GIS 13, pp. 47–61.*

Temple Lang, D. (2000) The Omegahat Environment: New Possibilities for Statistical Computing, Journal of Computational and Graphical Statistics 9, pp. 423–451, http://www.jstor.org/stable/1390938.

Temple Lang, D. (2007), R as a Web Client – the RCurl package, Journal of Statistical Software, http://cran.r-project.org/web/packages/RCurl/. www.omegahat.org/RCurl/RCurlJSS.pdf

Lang, D. (2010) XML: Tools for parsing and generating XML within R and S-plus', http://cran.r-project.org/package=XML

Vance, A. (2009) Data Analysts Captivated by R's Power, http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html, Last date accessed 2010-10-20.

Vretanos, P. A. (2005) OpenGIS Filter Encoding Implementation Specification 04-095, Technical Report, Open Geospatial Consortium.

Whiteside, A. (2007), OGC Implementation Specification 06-121r3: OGC Web Services Common Specification, Technical Report, Open Geospatial Consortium.

Socio-Spatial Modeling and Analysis

LandSpaCES: A Spatial Expert System for Land Consolidation

Demetris Demetriou[1], John Stillwell[1], Linda See1[1,2]

[1]School of Geography of Geography, University of Leeds, Leeds , UK
[2]Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria
demdeme@cytanet.com.cy, j.c.h.stillwell@leeds.ac.uk, see@iiasa.ac.at

Abstract. Land fragmentation is a major issue in many rural areas around the world, preventing rational agricultural production and sustainable rural development. Traditionally, land consolidation has been the primary land management approach for solving this problem. Land reallocation is recognised as the most important, complex, and time-consuming process of land consolidation. It is split into two components: land redistribution and land partitioning. In this paper, we outline a land redistribution model called *LandSpaCES* (Land Spatial Consolidation Expert System) which is the central module of LACONISS, a LAnd CONsolidation Integrated Support System for planning and decision making. *LandSpaCES* integrates GIS with an expert system (ES) and is able to generate alternative land redistributions under different scenarios. Two key system concepts are utilised: 'No-Inference Engine Theory (NIET),' which differentiates *LandSpaCES* from conventional ES development and a parcel priority index (PPI), which constitutes the basic measure that defines the redistribution of land in terms of location. The module has been applied to a case study area in Cyprus and the results compare very favourably against an independent solution derived previously by human experts.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_13,

1 Introduction

Land fragmentation refers to the situation in which landholdings consist of numerous spatially separated land parcels (Van Dijk 2003) that may be small in size, irregular in shape, dispersed from one another, and separated by many boundary lines. Frequently, land fragmentation implies a defective land tenure structure which may be preventing efficient agricultural production and sustainable rural development. Fragmentation is commonplace in rural areas around the world and agricultural censuses show that it has been a problem in Cyprus for several decades with average land holding size declining from 7.2 hectares in 1946 to 3.5 hectares in 2003 (Demetriou et al 2010a).

Land consolidation has been the most favoured land management approach for solving the land fragmentation problem and has been applied in many countries. It comprises two components: land reallocation (or land readjustment) and agrarian special planning (Thomas 2006). Whilst the former involves the rearrangement of the land tenure structure and is the core component of each land consolidation approach, the latter involves the provision of infrastructure such as roads, irrigation and drainage systems, landscaping, environmental management, village renewal, and soil conservation. The European Union (EU) has provided support for consolidation schemes from the European Agricultural Fund for Rural Development (EAFRD) and the Food and Agriculture Organisation (FAO), which has a long tradition of involvement in land consolidation activities. In Cyprus, land consolidation followed from a Land Consolidation Law enacted in 1969 and has been applied on a compulsory basis by resolution of the majority of the landowners concerned. Seventy three projects have now been completed covering an area of 17,552 hectares (Land Consolidation Department 2010) or 11.2% of the total agricultural area enumerated in the 2003 agricultural census. In addition, 15 projects are currently running and 34 projects are under study.

Although land consolidation has been successfully implemented in Cyprus since 1970, the process encounters major problems associated with land reallocation (e.g. the long duration of projects, high operational costs, conflicts of interest that arise among the stakeholders involved). At present, it is a planning and decision-making process that is not supported adequately by automated procedures. The problems with the current land reallocation process suggest the need for a new geotechnology tool to improve effectiveness in terms of the quality of planning and decision making, efficiency in terms of time and operational costs, transparency by structuring the process in a systematic and standardised form, expert

knowledge transfer, and the facilitation of training and automation whilst generating an optimal solution (Demetriou et al 2010a). Whilst efforts towards such systems have been ongoing since the 1960s, an effective solution to the problem has not yet been developed despite significant progress made in the development of geographic information systems (GIS). It is for these reasons that a new framework has been proposed for the development of LACONISS, an integrated planning and decision support system (IPDSS) for land consolidation that is summarised below and outlined in more detail in Demetriou et a. (2010b).

Land reallocation in LACONISS is split into two components: land redistribution and land partitioning. At the heart of LACONISS is a 'Design module' called *LandSpaCES*, which integrates a GIS with an ES. The advantages of this integration have been recognised by many researchers (e.g. Burrough 1986; Zhu and Healey 1992; Fischer 1994), and many spatial case studies have been carried out since 1980 dealing with issues such as site analysis, land-use planning, groundwater modelling and other spatial planning problems (e.g. Jun 2000; Kalogirou 2002; Filis et al, 2003; Yeh and Qiao 2004; Jin et al, 2006; McCarthy et al. 2008). The *LandSpaCES* module automatically generates a set of alternative land redistributions that include land parcel centroids (with full ownership details attached) representing the approximate location of the new parcels. In contrast to most existing studies that consider land redistribution as a mathematical optimisation problem, *LandSpaCES* treats land redistribution as a decision-making and evaluation problem, which is based on legislation, the existing land tenure structure, rules of thumb, and the knowledge and experience of the planner. The integration of GIS and ES has proved an effective solution to this problem.

The aim of the paper is to outline the design, development, and evaluation of *LandSpaCES*. A brief overview of LACONISS is provided in the next section. The design and development of *LandSpaCES* is then outlined in sections 3 and 4. Section 5 describes the case study utilised for the evaluation of *LandSpaCES*, which is described in section 6, together with an analysis of results for a set of 10 scenarios. Finally, section 7 ends with conclusions and a road map of future work on the development of LACONISS.

2 LACONISS: A framework for land consolidation planning

The operational framework of LACONISS, which is illustrated in Figure 1, is based on Simon's (1960) three-stage decision-making model: i.e. in-

telligence, design, and choice. It consists of three sub-systems: LandFragmentS (Land Fragmentation System) represents the 'Intelligence phase' of the process and involves building an appropriate GIS model and scanning the current land tenure system by utilising multi-attribute decision-making (MADM) methods to measure the extent of land fragmentation. Whereas MADM is a selection process between a discrete and limited number of alternative solutions which are described by criteria (Malczewski 1999), it will be used for measuring and exploring this multi-attribute problem using two extreme absolute values representing best and worst. *LandSpacES* contains (i) a design module that integrates the GIS with an ES and generates alternative land redistributions ('Design phase I') and (ii) an evaluation module that uses the GIS and MADM methods to evaluate alternative distributions and identify the most beneficial one ('Choice phase I'). The final output of *LandSpacES* is a map showing the centroids of parcels with their land value and ownership attributes, which are then transferred to LandParcelS (Land Parcelling System), the sub-system that creates the optimum set of boundaries (polygons) for the land parcels around each of the centroids by integrating the GIS with a genetic algorithm (GA) and multi-objective decision-making (MODM) methods ('Design and Choice phase II'). Our focus here is on *LandSpacES*.

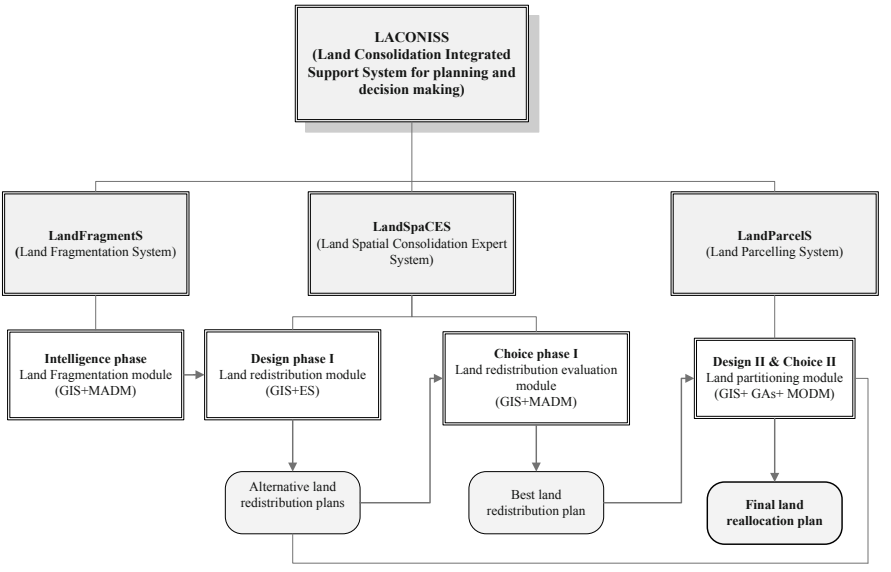


Fig. 1. The operational framework of LACONISS (Demetriou et al, 2010b).

3 *LandSpaCES* system design

LandSpaCES is a hybrid system that integrates an expert system (ES) with a GIS. An ES represents and reasons with knowledge, which is built to emulate the decision-making ability of a human expert. An ES typically consists of: a user interface which is responsible for the communication between the system and the user; a knowledge base which contains the knowledge about a problem domain; and an inference engine which carries out the reasoning for reaching a solution. The most popular knowledge representation technique uses rules which have the form of condition-action pairs, i.e. "IF this condition (or premise or antecedent) occurs, THEN some action (or results or conclusion, or consequence) will or should occur."

The design of an ES or knowledge-based system involves tasks that are different to those found in the development of conventional software systems as a result of the knowledge component. The design task usually consists of two main components: knowledge definition and knowledge detailed design (Giarratano and Riley 2005). For the purposes of this research, the design task is split into the following steps: system definition, knowledge acquisition, knowledge representation, knowledge-base building, and the definition of the inputs and outputs. Each of these steps is now considered in more detail.

3.1 System definition

The first step is to define the objectives of the system. In the case of the 'Design module' of *LandSpaCES*, the objectives are to automate the process of land redistribution so as to generate a complete problem solution; to be used as a decision support tool by generating alternative land redistributions; to enhance the land redistribution process by structuring it in a systematic, standardised, and transparent way using an appropriate model; and to considerably diminish the time needed by a human expert to carry out the process. In addition, the results of *LandSpaCES* can be used as inputs to the *ex ante* evaluation of a land consolidation project based on EU requirements (Demetriou et al*,* 2010a) to test the implications of various scenarios. Furthermore, the system may also be used as a trainee tool for new and expert land consolidation technicians to understand and analyse the reasoning process of land redistribution. The main stakeholders involved in the land consolidation process, i.e. landowners,do not participate

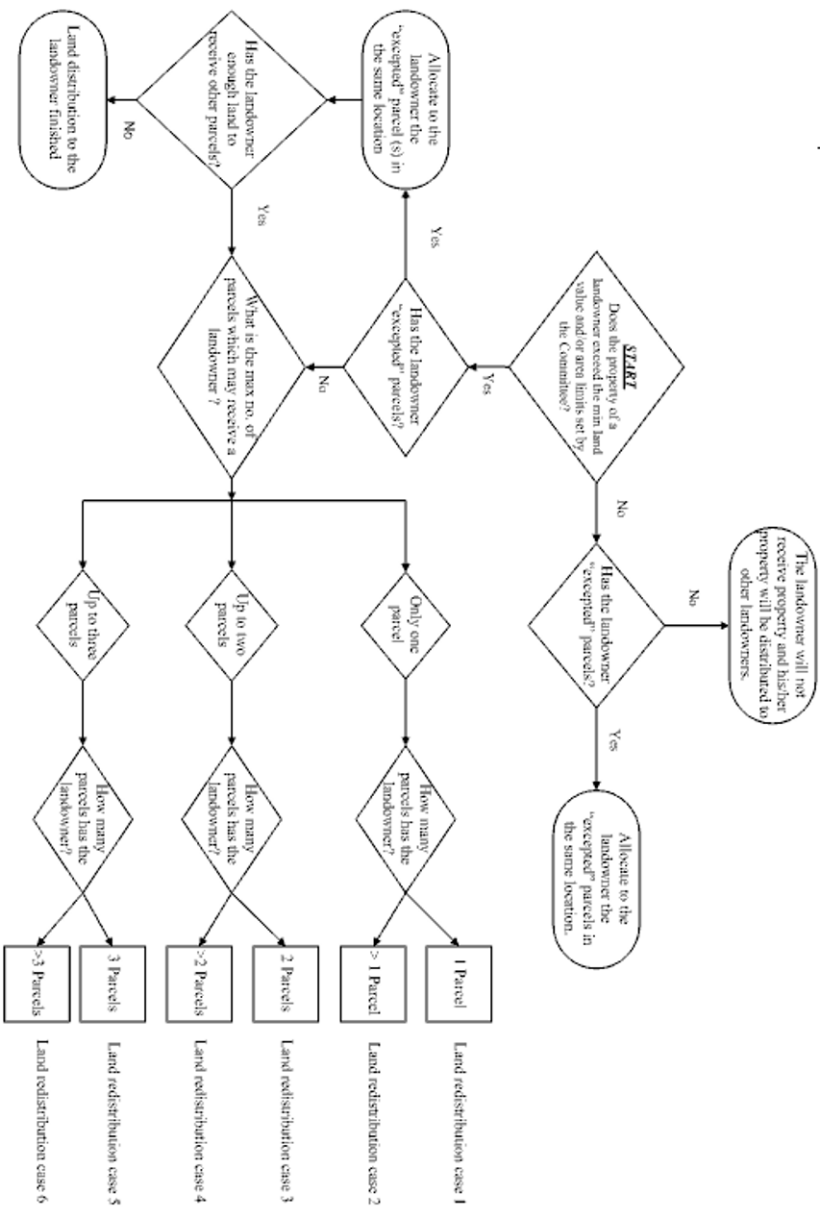


Fig. 2. The main decision tree for the Design module of *LandSpacES*

in this automated process since their preferences are predicted in some way, as we will see later, using a parcel priority index (PPI).

However, a crucial matter which will be considered by the authors in a later stage of the research is the involvement of stakeholders in the system evaluation process.

3.2 Knowledge acquisition

The second stage in the design of an ES involves knowledge acquisition, which Liou (1998, p.2-1) defines as *"the process of extracting, structuring and organizing knowledge from several knowledge sources, usually human experts, so that the problem solving expertise can be captured and transformed into computer readable form."* A variety of knowledge acquisition methods have been suggested; the methods tend to be a combination of approaches and sources including documentation (e.g. manuals, guidelines and legislation); studying past projects and their subsequent shortcomings; discussing cases with experts via personal or collaborative interviews; and observing experts applying their knowledge to current problems.

The selection of an appropriate knowledge acquisition method depends on many factors, namely, the problem domain, the availability of knowledge resources, and the time/cost constraints. In this research, knowledge was collected through the following sources: the principal author's thirteen-year personal experience of working on land consolidation projects; informal discussions/interviews with expert land consolidation technicians (i.e. the prospective main users of the system); documentation, such as Land Consolidation Law, formal LCD guidelines and instructions, legal advice, etc; and an analysis of the solution given by experts in the case study used in this research.

3.3 Knowledge representation

The third fase in the design of an ES involves determining a mechanism for how the knowledge acquired will be represented. To represent the decisions that compromise the decision making process, decision trees have been utilised. The problem has been split into seven sub-problems represented by decision trees. The main decision tree shown in Figure 2 ends in six different types or cases of land redistribution, where each case is further represented by a separate decision tree. The main decision tree represents the general decisions taken by the experts regarding land redistribution. For example, one of the first decisions is whether a landowner will receive property in the new plan. If the answer is yes, then the next decision involves determining the maximum number of parcels that may be al-

located to the landowner in the new plan. Based on the answer to these decisions, the landowners are then classified into six different types of land redistribution, and a further decision tree is used for each case to determine how to reallocate the property of each landowner in terms of the number and the location of new parcels.

3.4 Building the knowledge base

From the decision trees, 74 IF-THEN rules have been extracted. There are 22 generic rules which relate to the main decision tree and determine the high level outcomes, i.e. whether a landowner receives property in the new plan and, if relevant, the maximum number of parcels that can be allocated to each landowner. The six decision trees that correspond to the land redistribution types (or cases) contain 38 rules and focus on the decisions specific to each land redistribution case, e.g. whether to create a new parcel or whether to move a new parcel to another location.

The rules are grouped into 'rule clusters' that represent the decision-making process for a certain land redistribution sub-problem, which in turn is derived by a decision tree. Decision trees and their rules follow a 'forward chaining' (i.e. data driven) inference approach for solving the problem. The rules execute based on the input data and 'facts' in order to reach the conclusions. Two examples of rule clusters, one for the main tree and one for the land redistribution Case 2, are presented in Tables 1 and 2, respectively. The complete set of decision trees and the relevant code for the rules shown in Tables 1 and 2 can be found in Demetriou *et al.* (2010c). It is noted that these rules lack a spatial dimension apart from the location of each new parcel and they focus on the decision making part of land reallocation whereas another system module called LandParcelS is used to establish the spatial arrangement of the new parcels.

3.5 Definition of the inputs and outputs

The final step in the design of an ES is defining the inputs and outputs to the system. This particular problem requires two kinds of inputs, i.e. spatial data and 'facts'. The spatial data are cadastral layers and their attribute tables from a GIS as well as additional related database tables. Facts are decision variables that are input by the user, which are used by the rules to infer new parameters or conclusions or actions. Changing the facts will result in alternative land redistribution solutions. The system outputs are a database table or a map and indicate: (i) those landowners taking property in the new plan and those that do not; (ii) the total area and land value of

the property that each landowner receives in the new plan; (iii) the number of parcels that each landowner receives in the new plan; (iv) the area and land value of each new parcel; and (v) the approximate location (i.e. centroid) of the new parcel(s) by landowner. The next section describes how the design is turned into a working system.

4 *LandSpaCES* system development

4.1 Development tools

A literature review of ES development (e.g. Choi and Usery 2004; Giarratano and Riley 2005; Hicks 2007) suggests that the easiest and most efficient way to develop an ES is to use a specialised tool, i.e. an ES shell, a knowledge engineering tool or an artificial intelligence language. However, these specialised tools are designed for the development of standalone ES applications and not hybrid systems, e.g. involving GIS, since they have limitations in their flexibility and integration with non-ES applications (Jin et al*,* 2006). Unfortunately, there is a lack of specialised ES development tools capable of easily integrating ES into proprietary GIS, despite the fact that knowledge and expertise are important components of all spatial planning processes. A solution to this problem is to use a conventional programming language that provides greater development flexibility even though it is a time-consuming task for developing an ES from scratch (Lukasheh et al*,* 2001).

Thus, it was decided to sacrifice some of the advantages provided by ES development tools to gain the advantages offered by conventional programming languages under a common GIS development environment. Therefore, VBA and ArcObjects in ArcGIS were used to fully integrate GIS with the ES component. However, this integration must be based on robust theoretical foundations. This requirement is provided by the 'No-Inference Engine Theory' (NIET) as described in the next section.

Table 1. Rules in the main cluster

Rule No.	IF	THEN
1	The total area **OR** value of a landowner's property **<** the corresponding minimum completion limits set by the Committee **AND** the examined parcel is not "excluded"	The landowner will not receive any parcel in the new plan **AND** he will receive expiate the land value of the property **AND** the property will be available to be distributed to others
2	The total area **OR** value of a landowner's property >= the corresponding minimum completion limits set by the Committee **AND** the parcel examined is not "exclusive" AND the landowner has not applied to be completed	The landowner will not receive any parcel in the new plan **AND** he will receive expiate the land value of the property **AND** the property will be available to be distributed to others
3	The total area **OR** value of a landowner's property **>=** the corresponding minimum completion limits set by the Committee **AND** the total area is >= minimum area limit set by the Law	The landowner will receive property in the new plan
4	The total area of a landowner's property **>=** the corresponding minimum completion limit set by the Committee **AND** the total area is <= minimum area limit set by the Law and the landowner has applied "to be completed"	The landowner will receive property in the new plan
5	The total value of a landowner's property **>=** the corresponding minimum completion limit set by the Committee **AND** the total area is <= minimum area limit set by the Law and the landowner has applied "to be completed"	The landowner will receive property in the new plan
6	The total area of a landowner's property **>=** the corresponding minimum completion limit set by the Committee **AND** the total area is <= minimum area limit set by the Law and the landowner has not applied "to be completed"	The landowner will not receive any parcel in the new plan **AND** he will receive expiate the land value of the property **AND** the property will be available to be distributed to others
7	The total value of a landowner's property **>=** the corresponding minimum completion limit set by the Committee **AND** the total area is <= minimum area limit set by the Law and the landowner has applied not "to be completed"	The landowner will not receive any parcel in the new plan **AND** he will receive compensation equal to the land value of the property **AND** the property will be available to be distributed to others
8	A landowner will not receive property in the new plan	His property will be available to be distributed to others

Table 2 Rules for the rule cluster for land redistribution Case 2.

Rule No.	IF	THEN
1	The new area will be allocated to a landowner **<** the minimum area limit set by the Law	The new area will be equal to the minimum area limit set by the Law
2	The area of the new parcel <= available land in the block of that parcel	Create the new parcel
3	The number of parcels already allocated to a landowner = the maximum number of parcels may received by the certain landowner	Do not allocate him any other parcels
4	The area of the new parcel > the available area of the block of the parcel **AND** the Parcel Priority Index > the minimum of the parcels of that block	Search and allocate the examined parcel in that block and "move" the parcel(s) with less PPI in another block
5	*Rule 4 can not be satisfied*	Search and allocate the examined parcel in another block in which the landowner posses a parcel
6	*Rule 5 can not be satisfied*	Allocate the new parcel in none block to decide the user

4.2 No-Inference Engine Theory

NIET was proposed by Hicks (2007) although the concept has been employed by other researchers (e.g. Griffin and Lewis 1989). The basic feature of NIET is that the knowledge base and inference engine are not kept separate as in conventional ES. In NIET, there is no inference engine and the reasoning (rule base) and process (inference engine) are combined into a single unit. This transforms the traditional inference engine into a procedural solution involving a sequence of IF-THEN statements. Thus, the rules are ordered in a logical sequence during the development stage. In the situation where two or more rules have at least the first condition of their premise in common, the conflict is resolved by firing the rules with the greater number of conditions, so that they can be tested first. This conflict resolution strategy is commonly employed and is the default for most ES products. Another feature of NIET is that rules can be grouped into 'rule clusters' (capturing sub-problems) depending on the task which is an efficient programming technique for breaking down complex problems in a manageable and understandable size.

4.3 The parcel priority index

Crucial matters to be considered when building the land redistribution model are the way in which the preferences of the landowners are incorporated and the need to ensure equity, transparency, and standardisation of the process in terms of the location and the allocation of the new parcels. Regarding the former, it is accepted that the most important concern for landowners in a land consolidation project is the location of the new parcels that they will receive. It is also well known by land consolidation planners that each landowner wishes to receive property in the location of their 'best parcel' then the next 'best parcel' and so on (Sonnenberg 1998; Cay et al, 2006; Ayranci 2007). Practice has shown that the 'best parcel' is perceived as that with the largest area and/or the highest land value per hectare (either market price or agronomic value) or a combination of these two factors. The parcel priority index (PPI) has been devised to take into account both factors; it is a proxy for the preference of the landowners and a determinant of the location and the allocation of each new parcel. The PPI, which may take values from 0 to 1, is calculated for land area as follows:

$$PPI(A_i) = \frac{(A_i - MinA) * 0.5}{MeanA - MinA} \quad (\text{if } A_i \le MeanA) \tag{1}$$

$$PPI(A_i) = \frac{(A_i - MeanA) * 0.5}{MaxA - MeanA} + 0.5 \quad (\text{if } A_i > MeanA) \tag{2}$$

where $PPI(A_i)_i$ is the PPI based on the area of parcel i, A_i is the area of parcel i, and $MinA, MaxA, MeanA$ are the corresponding area values for all the parcels in the dataset.

The PPI for land value is calculated as:

$$PPI(V_i) = \frac{(V_i - MinV) * 0.5}{MeanV - MinV} \quad (\text{if } V_i \le MeanV) \tag{3}$$

$$PPI(V_i) = \frac{(V_i - MeanV) * 0.5}{MaxV - MeanV} + 0.5 \quad (\text{if } V_i > MeanV) \tag{4}$$

where $PPI(V_i)$ is the PPI based on the land value of parcel i, V_i is the land value of parcel i, and $MinV, MaxV, MeanV$ are the corresponding land values for all the parcels in the dataset.

The overall PPI_i for parcel i is then defined as:

$$PPI_i = W_A * PPI(A_i) + W_V * PPI(V_i) \tag{5}$$

where W_A, W_V are the weights for area and land value respectively that should sum to 1.

Initially, the PPI is calculated separately for each parcel based on a linear scale transformation based on the assumption that the minimum, maximum, and mean values of a dataset correspond to scores 0, 1, and 0.5, respectively. The mean is involved in the transformation in order to avoid great variations between the PPI values when extreme values of the factors involved are present. Thereafter, the overall PPI is calculated based on the relevant weight assigned by the user for each factor. Weights represent the importance of each factor in the land redistribution process and influence the location-allocation of the new parcels. The overall PPI, results in the ranking of all the parcels in a project which defines the priority of a land-

owner-parcel pair in terms of allocating a parcel in a certain location or not. The potential use of median instead of mean would have the following influence: if the median is less than the mean for the area and land value data, then the PPI values for each parcel will be increased in favour of the smaller than the median area and land values. In contrast, if the median is larger than the mean for the area and land value data, then the PPI values for each parcel will be increased in favour of the larger than the median area and land values. However, land redistribution results will not be influenced since the relative variation of PPI from parcel to parcel will be proportional.

In addition, the overall PPI is utilized for ranking the parcels of the holding of each landowner defining location preferences for the landowner's new parcels. In other words, the parcel of a landowner with the greatest PPI represents the first preference of the landowner in terms of allocation. However, it is realized that sometimes it will not be possible to satisfy the highest preferences of all landowners because land will not be available in some blocks. Thus, in such cases, there is a conflict among the landowners' preferences. This conflict is solved by employing the initial ranking of all parcels based on the overall PPI, which defines the priority of a landowner-parcel pair in the land redistribution process in terms of allocating a parcel in a certain location or not. The higher the PPI, the higher the priority, hence the higher possibility for a landowner to receive his property in the desired location(s).When the land redistribution process is in progress, not every parcel-landowner pair is ensured a location in the new plan (and they may be 'displaced' at any time during the process) until the land redistribution process has terminated. Ideally, the PPI may take into account further reallocation criteria regarding a parcel and a landowner. For example, other criteria could be land use, morphology, landowner's profession (farmer or not farmer, full-time or part-time farmer), and age or residence (in the village or not).

5 A case study

The case study for implementation and evaluation of *LandSpaCES* was selected based on the following criteria: the availability of the data in a computerised form; a reasonable volume of data; a comprehensive representation of the problem domain so as to reflect as many cases of land redistribution as possible; and the independence of the solution, i.e. the relevant land reallocation solution that was made by a group of experts, will be completely independent from the person who has developed the

system. Based on reviewing a number of different completed land consolidation projects in Cyprus, a case study covering the land consolidation of the village of Chlorakas was selected.

Chlorakas is located in the District of Pafos, at an altitude of 70m above mean sea level and at a distance of 3km to the north of the town of Pafos. The location of the case study village on a map of Cyprus is shown in Figure 3. The village administrative boundaries cover a total area of 492 hectares of lowland while the extent of the consolidated area is 195 hectares. The main types of cultivation in this area are citrus fruits, grapes, vegetables, and bananas. This project was amongst the first applied in Cyprus. It began in March 1971 and was completed in June 1974. A cadastral map showing the layout of the parcels, roads, etc. before and after land consolidation is shown in Figure 4.

For the purpose of this research, the following data were provided by the Land Consolidation Department (LCD) regarding the study area before and after project implementation: cadastral databases containing attribute information about landowners, original and new parcels, etc; cadastral maps at scales of 1:5,000 and 1:2,500 (Figure 4); and documents, such as catalogues with cadastral information, proceedings of the meetings of the Land Consolidation Committee (LCC). Based on the data collected, a geodatabase was created consisting of two datasets containing the information before (original data) and after land consolidation (the human experts' solution). The human experts are the land consolidation technicians who usually work in groups of two and who initially prepare a land reallocation plan for a certain case study. However, the latter is inspected regularly by a team in the District Office and a team in the Head Office. Then, the final land reallocation plan must be approved firstly by the Head of the Department and finally by the LCC.

Dataset 1 consists of three layers and two database tables. It is used as input to the system to create its outputs, i.e. output layers and database tables. Dataset 2 consists of one layer and two databases tables. It is also used as an input to the system for evaluation. A relationship was established among layers and database tables for each dataset. The quality of both data sets was considered in terms of completeness, i.e. the lack of errors of omission in a database; compatibility, i.e. the ability of different datasets or different layers and databases to be combined to produce outputs; consistency, i.e. the absence of apparent contradictions in a database; applicability, i.e. the appropriateness or suitability of data for a set of commands, operations or analyses; accuracy that is the inverse of error and concerns spatial and attribute data; precision, i.e. the recorded level of detail of a dataset; bias, i.e. the systematic variation of data from reality; and resolution.

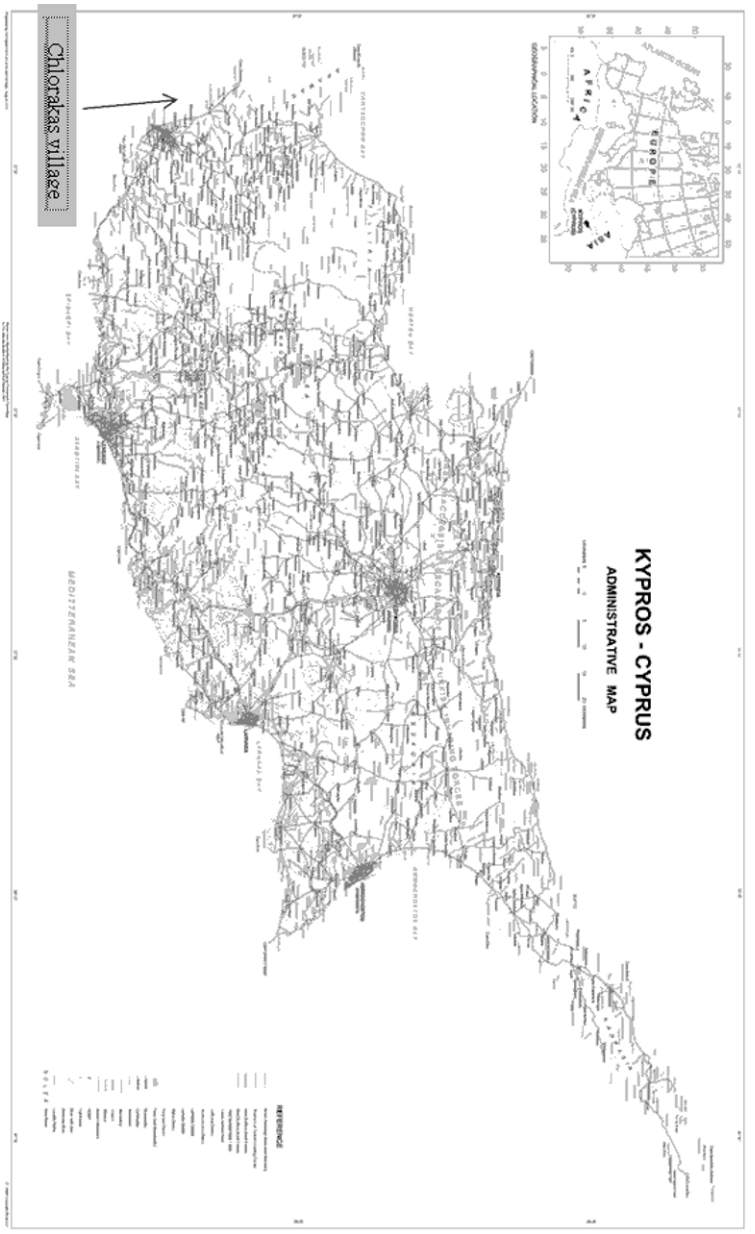


Fig. 3. Location of the case study village (provided by the Land Surveys Department of Cyprus)

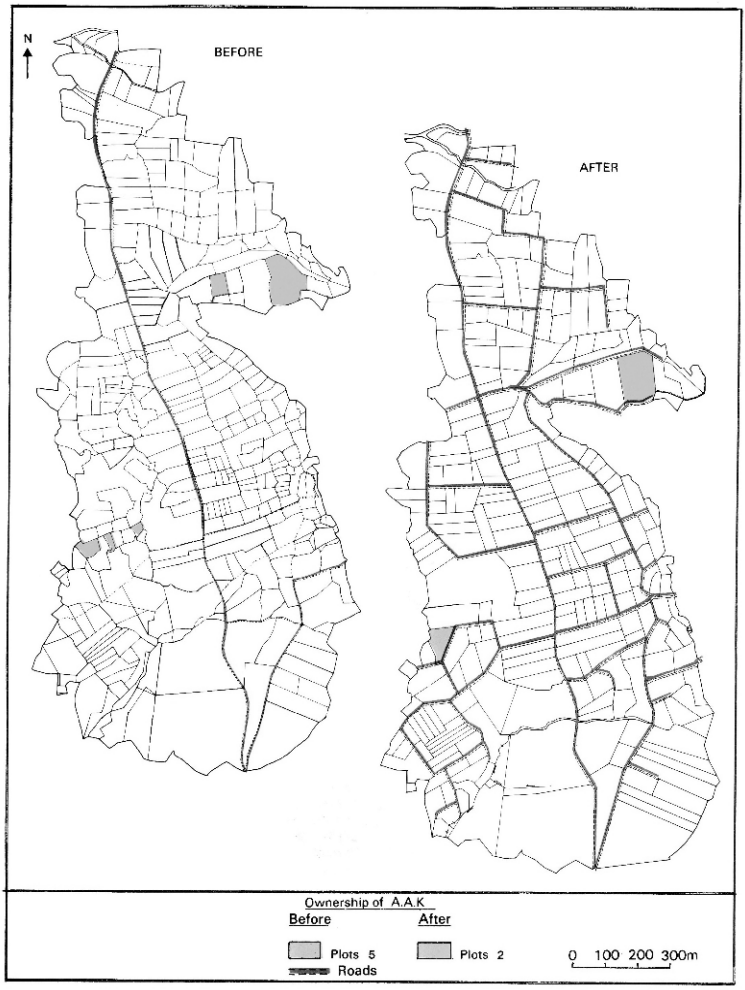


Fig. 4 Study area before and after land consolidation (LCD 1993)

6 System evaluation

The evaluation of an ES involves the processes of verification (building a system that functions correctly in terms of eliminating errors) and validation (building a system that operates as it should in terms of the quality of the decisions made) (O'Keefe et al, 1987). Although *LandSpaCES* has

been subjected to both processes, the focus here is on validation and therefore how well the system performs.

6.1 Validation

Based on the literature review (e.g. O'Keefe et al, 1987; Sojda 2006), the 'establishing criteria' approach and the 'test case' method have been selected as the most appropriate way to validate the system. Using the 'Chlorakas' case study as an input, the basic output of *LandSpaCES* is the location of the centroid of each new parcel as shown in Figure 5. The fraction above each dot indicates the Parcel_ID and Owner_ID of that new parcel. Another output provided by *LandSpaCES* is a database containing the attributes of each parcel, i.e. the size, land value, landowner, etc. The double lines represent roads that are defined by the planner. The underlined numbers represent Block-IDs, where a block is a sub-area enclosed by roads, physical boundaries (e.g. a stream or a river), and/or the external boundaries of the case study area.

Establishing validation criteria

The system performance is measured by comparing the agreement between the decisions made by the system and those taken by the human experts. Although the decisions made by the human experts do not necessarily result in the optimal solution (due to the manual processing of vast amounts of information and a degree of subjectivity in the decision making), this is the standard way of evaluating ES since the aim of such a system is to emulate the human reasoning process utilised for solving a narrow problem domain. Nine evaluation criteria were used to evaluate system performance as shown in Table 3. These criteria cover the most important decisions made by the expert regarding the land redistribution plan.

Evaluation of system performance

Table 3 shows the system performance for each validation criterion (CR1-CR9) while Table 4 shows the system performance for each land redistribution group. It is noted that a 'completed parcel' is a new parcel that is allocated to a landowner when the total size of holding is less than the minimum size provided by legislation for the new parcels in a certain land consolidation area. The term 'new parcels created' refers to each new parcel that has attributes stored in a database table and its approximate location represented as a point in a layer. The complete geometrical design in

terms of final location and shape, which constitute the topology of each new parcel, is an issue that will be considered in the development of the LandParcelS module.

The results of the system performance are very encouraging since the system reproduces the human expert decisions with an agreement of between 62.6 to 100% for the nine validation criteria. For criteria 1, 2, 4 and 5, the agreement is 100% or close to perfect. For criteria 3, 6, and 7, the agreement is over 70%. The lower performance for these criteria is due to the fact that the system is not yet able to directly model certain kinds of information such as: the actual landowners' preferences and demands and the pre-decisions of the planning team which may have resulted in a decision that violates a relevant legislation provision (based on justification).

Table 3. *LandSpaCES* performance based on nine validation criteria

Validation criterion	LandSpaCES	Human Expert	System performance %
Number of landowners received property (CR1)	210	204	98.04
Number of the common landowners received property (CR2)	Agreement in 204 out of 204 landowners		100.00
Number of landowners received a "completed" parcel (CR3)	31	24	70.83
Number of common landowners received a "completed" parcel (CR4)	Agreement in 24 out of 24 landowners		100.00
Total number of new parcels created (CR5)	268	267	99.63
Number of the new parcels created per owners' group (CR6)	See details in Table 5 below		69.23-100
Number of the new parcels received by each landowner (CR7)	Agreement in 219 out of 253 landowners		86.56
Number of new parcels received by each landowner in common blocks (CR8)	Agreement in 210 out of 267 new parcels		78.65
Number of new parcels received by each land owner in a common location (CR9)	Agreement in 167 out of 267 new parcels (See Figure 6 below)		62.55

Table 4. Number of new parcels created per land redistribution group

Land redistribution group	LandSpaCES	Human Expert	Difference	System Performance (%)
Completed parcel	31	24	+7	70.83
Case 1	80	79	+1	98.73
Case 2	26	26	0	100.00
Case 3*a	32	32	0	100.00
Case 3b	18	16	+2	87.50
Case 4	18	26	-8	69.23
Case 5	26	24	+2	91.67
Case 6	37	38	-1	97.37
Total	268	265+2**=267		

* Case 3 has been programmatically split into two parts (a and b) for efficiency reasons

**Human expert allocated 2 exempted parcels from rules that are not classified in the land distribution groups

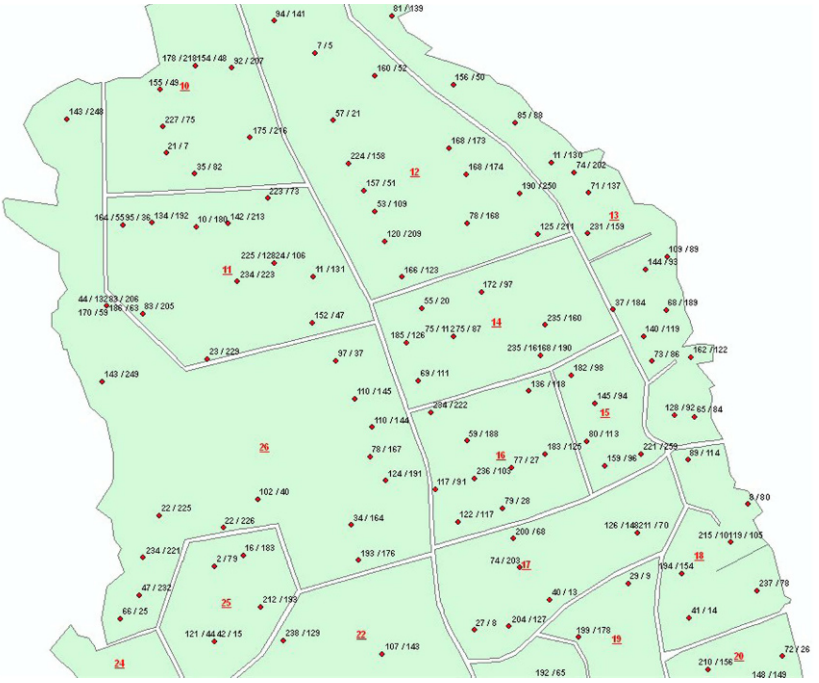


Fig. 5. The final system output (a part of the case study area)

Criteria 8 and 9 are associated with the location of the new parcels. Figure 6 shows the location of each parcel centroid, which will be the

starting point for the generation of parcel boundaries in the LandParcelS module. Figure 6 also shows the actual parcel boundaries that were designed by the experts in the case study. We will eventually compare these boundaries with the output from LandParcelS in a future paper. As mentioned in section 4.4, the system defines the location of each new parcel based on the PPI. Although the percentage of agreement between the system and the experts is quite high (78.7% and 62.6% for criteria 8 and 9, respectively), the predictions will naturally differ since the experts used the actual landowners' preferences to determine the locations of the new parcels. In addition, some differences may be due to the fact that planners also try to satisfy landowners' preferences against the effectiveness of the plan while the system follows an objective, transparent, and systematic process to determine the location and allocation of the new parcels.

In terms of the operational system performance, the system considerably outperforms the human experts in terms of the amount of time taken to complete the process. A small survey carried out based on 10 expert land consolidation technicians showed that an individual expert needs about 30 working days to solve this particular land redistribution problem whilst *LandSpacES* needed only 6 minutes, which is an impressive reduction in time for this task.

Overall, the results showed that the system performance compared to the human experts' solution is very good, but further improvements could still be made by adding more rules to the knowledge base. In addition, in a further system enhancement, the direct incorporation of extra data (e.g. the actual landowners' preferences, land use, the landowners' personal data, i.e. residence, age, occupation, etc.) needs to be considered. Also, the system suffers from the well-known ES deficiency regarding the ability to produce new knowledge through experience from previous results such as provided by machine-learning techniques (e.g. neural networks and genetic algorithms). However, such techniques are not appropriate for classical decision-making problem domains based on human reasoning (Openshaw 1997; Negnevitsky 2002; Padhy 2005). Eventually, testing the system with more case studies may also extract more robust conclusions regarding its performance.

Fig. 6. The parcels' centroids allocated by the system (large dots) which fall within the boundaries of the same parcels allocated by experts (a part of the case study area).

6.2 Generation of alternative solutions

The primary aim of the Design module of *LandSpaCES* is to be able to generate alternative land redistributions by changing the input 'facts' or what are essentially different scenarios. Thus, the system was run with 10 different sets of facts to generate 10 alternative land redistributions in order to demonstrate that the system works properly for a range of facts. In particular, the interaction between the facts and the results was tested in terms of the logic of the outcomes produced and the variation in the results. The input 'facts,' which involve 11 different land redistribution variables, and the results for three main land redistribution criteria, are shown

in Table 5. It should be noted that the values of the facts must be feasible with respect to a particular project; otherwise the results will also be unfeasible and unrealistic.

A general picture derived from Table 5 is that changing the facts can generate quite different solutions. In particular, some remarkable but expected findings are the following: Fact F1 is crucial for outputs C1 and C3; Facts F2 and F3 strongly influence the results of the three basic outputs, i.e. C1, C2 and C3; Facts F4 to F8 affect only output C1; Facts F9 and F10, which represent the PPI weights, do not cause any change in the three outputs because the PPI influences only the location-allocation (that is not represented in the three outputs) of the new parcels; and a dramatic change in all the facts except in the case of F9 and F10 cause a substantial change in all the outputs. Further analysis of the interaction between the facts and the outputs is carried out in Demetriou *et al.* (2010c). The above results indicate that the system is reliable in generating various alternative land redistributions by using different sets of facts. These solutions can then be passed to the Evaluation module of *LandSpacES* for assessment using MADM methods and more than the three criteria used in this section. The Evaluation module is currently under development.

Table 5. Facts and outputs for ten alternative land redistributions

Alternative	Facts											Outputs/Criteria		
No	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	C1	C2	C3
A1	2676 (2)	1000	100	2676 (2)	5351 (4)	5352 (4)	10702 (8)	10703 (8)	0.5	0.5	2000	268	210	31
A2	2676(2)	1500	150	2676 (2)	5351 (4)	5352 (4)	10702 (8)	10703 (8)	0.5	0.5	2000	264	206	27
A3	2676 (2)	2000	200	2676 (2)	5351 (4)	5352 (4)	10702 (8)	10703 (8)	0.5	0.5	2000	259	201	22
A4	2676 (2)	1000	100	2676 (2)	5351 (4)	5352 (4)	10702 (8)	10703 (8)	0.3	0.7	2000	267	210	31
A5	2676 (2)	1000	100	1338 (1)	4014 (3)	4015 (3)	6690 (5)	6691 (5)	0.5	0.5	2000	282	210	31
A6	1338 (1)	750	75	1338 (1)	4014 (3)	4015 (3)	6690 (5)	6691 (5)	0.5	0.5	1000	302	210	9
A7	1338 (1)	500	50	1338 (1)	4014 (3)	4015 (3)	6690 (5)	6691 (5)	0.5	0.5	1000	305	213	12
A8	1338 (1)	500	50	1338 (1)	2676 (2)	2677 (2)	5352 (4)	5353 (4)	0.5	0.5	1000	314	213	12
A9	2676 (2)	1000	100	2676 (2)	5351 (4)	5352 (4)	10702 (8)	10703 (8)	0.7	0.3	2000	268	210	31
A10	6690 (5)	4000	350	6690 (5)	9366 (7)	9367 (7)	14718 (11)	14719 (11)	0.5	0.5	4500	204	182	89

Facts

F1 The minimum parcel area limit (in m^2) for this land consolidation area as set by legislation
F2 The minimum holding's size limit (in m^2) for a landowner to receive a parcel in the newplan as set by the Committee
F3 The minimum holding's land value limit (in CyP) for a landowner to receive a parcel in the newplan as set by the Committee
F4 The lower limit (in m^2) of a "small" holding size
F5 The upper limit (in m^2) of a "small" holding size
F6 The lower limit (in m^2) of a "medium" holding size
F7 The upper limit (in m^2) of a "medium" holding size
F8 The lower limit (in m^2) of a "large" holding size
F9 The weight for parcel area for the calculation of the PPI (Parcel Priority Index)
F10 The weight for parcel land value for the calculation of the PPI (Parcel Priority Index)
F11 The minimum residual area limit (in m^2) for the creation of a newparcel for those landowners may receive more than one
Note: the number in brackets represents the area in donums (1 donum =1338 m^2)

Outputs/Criteria

C1 Total number of newparcels created
C2 Number of landowners received property in the newplan
C3 Number of landowners "completed"

Note: the number in brackets represent the area (1 donum=1,337.78m^2)

7 Conclusions

This paper has outlined a spatial expert system called *LandSpaCES*, which is the basic module of an IPDSS for land consolidation called LACONISS. *LandSpaCES* is capable of automatically generating a set of alternative land redistributions for different scenarios. The module has been applied to a case study area in Cyprus and the results showed a high system performance in terms of replicating an independent solution derived previously by human experts and an impressive performance in terms of time. The results clearly demonstrate that land redistribution is a semi-structured decision-making problem rather than an optimisation problem as considered in most existing studies as argued by Demetriou et al. (2010b).

Furthermore, *LandSpaCES* has transformed land redistribution into a systematic, transparent and effective process compared to the way in which it is currently carried out, i.e. in a manual, slow, semi-computerised manner. Thus, in general, the successful integration of GIS with ES proved that the latter technology, despite its decline since the 1990s, is still valuable for solving complex spatial planning problems that involve decision making. There were real challenges faced in solving this problem because of the lack of specialised tools for integrating GIS and ES. So there is a current technology gap since spatial planning problems inherently involve decisions based on heuristics, which renders these problems particularly suitable for ES.

Moreover, the two key system concepts, i.e. NIET and PPI, are both innovations in this problem domain. In particular, it has been demonstrated that NIET is an efficient alternative way for building an ES and fully integrating it within a GIS; despite some limitations (e.g. the knowledge base is not easily edited and an explanation facility is not provided) that are a result of not using specific ES tools for development. The PPI has been devised specifically for solving this problem, i.e. how to redistribute the land in terms of location. It is a proxy for the preferences of landowners and ensures the equity and transparency of the land redistribution process.

Although *LandSpaCES* has some limitations that could be tackled by adding more rules to the knowledge base or incorporating additional data into the model, it is a valuable contribution to solving the land redistribution process in terms of automation, effectiveness, equity, and transparency, which has potential applicability to many other countries once the relevant country-specific knowledge base is developed.

References

Ayranci, Y. (2007) Re-allocation aspects in land consolidation: A new model and its application. Asian Network for Scientific Information. *Journal of Agronomy*, 6(2), 270-277.

Burrough, P. A. (1986) *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford University Press.

Cay, T. and Iscan, F. (2006) Optimisation in land consolidation. Shaping the change. XXIII FIG Congress, Munich, Germany, October 8-13.

Choi, J. and Usery, E. (2004) System integration of GIS and a rule-based expert system for urban mapping. *Photogrammetric Engineering and Remote Sensing*, 70(2), 217-224.

Demetriou, D., Stillwell, J. and See, L. (2010a) Land consolidation in Cyprus: Why is an integrated planning and decision support system required? *Land Use Policy* submitted.

Demetriou, D., Stillwell, J. and See, L. (2010b) A framework for developing an integrated planning and decision support system for land consolidation? *Environment and Planning B: Planning and Design* submitted.

Demetriou, D., Stillwell, J. and See, L. (2010c) *LandSpacES:* A design module for land consolidation: Methods and application. *Working Paper 10/08*. School of Geography, University of Leeds, Leeds. http://www.geog.leeds.ac.uk/research/wpapers.

Filis, I., Sabrakos, M., Yialouris, C., Sideris, A. and Mahaman, B. (2003) GEDAS: an integrated geographical expert database system, *Expert Systems with Applications* 24 25-34.

Fischer, M.M. (1994) From conventional to knowledge-based geographic information systems. *Computers, Environment and Urban Systems*, 18(4), 233-4.

Giarratano, J. and Riley, G. (2005) *Expert Systems: Principles and Programming*. Course Technology. Canada.

Griffin, N., Lewis, F., 1989. A rule-based inference engine which is optimal and VLSI implementable. Tools for Artificial Intelligence. Architectures, Languages and Algorithms. IEEE International Workshop, 23-25 Oct, pp 246 – 251

Heywood, I., Cornelius, S. and Carver, S., (2002) *An Introduction to Geographical Information Systems*. 2nd ed., Pearson Education Limited, Harlow.

Hicks, R., (2007) The no inference engine theory-Performing conflict resolution during development. *Decision Support Systems*, 43(2), 435-444. http://www.sli.unimelb.edu.au/fig7/Brighton98/Comm7Papers/TS30-Rosman.html

Jin, Z., Sieker, F., Bandermann, S., Sieker, H. (2006) Development of a GIS-based Expert System for on-site storm water management. *Water Practice & Technology*, 1 1.

Jun, C. (2000) Design of an intelligent Geographical Information System for multi-criteria site analysis. *URISA Journal*, 12(3), 5-17.

Kalogirou, S. (2002) Expert systems and GIS: an application of land suitability

evaluation. *Computers, Environment and Urban Systems*, 26, 89-112.
Land Consolidation Department (LCD), (2010) Annual Report (2009) for land consolidation. Nicosia, Cyprus.
Land Consolidation Department (LCD), (1993) Land Consolidation in Cyprus. Ministry of Agriculture and Natural Resources of Cyprus Nicosia.
Liou, Y. (1998) Expert system technology: knowledge acquisition. In: Liebowitz, J., (eds.) 1998. *The Handbook of Applied Expert Systems*. CRC Press, MA, USA, 2-1 - 2-11.
Lukasheh, A., Droste, R. and Warith, M. (2001) Review of ES, GIS, DSS and their applications in landfill design management. *Waste Management and Research*, 19, 177-185.
McCarthy, J., Graniero, P. and Rozic, S. (2008) An integrated GIS-Expert System framework for live hazard monitoring and detection. *Sensors*, 8(2), 830-846.
Negnevitsky, M. (2002) *Artificial Intelligence: A Guide to Intelligent Systems*. Second edition. Addison Wesley, Essex.
O' Keefe, M., Balci, O. and Smith, E. (1987) Validating expert system performance. *IEEE Expert,* 2(4), 81-89.
Openshaw, S. and Openshaw, C. (1997) *Artificial Intelligence in Geography*. Wiley, West Sussex, pp 329.
Padhy, N. (2005) *Artificial Intelligence and Intelligent Sys*tems. Oxford University Press, New Delhi.
Simon, H. (1960) *The New Science of Management Decision*. Harper & Row, New York.
Sojda, R. (2006) Empirical evaluation of decision support systems. *Environmental Modelling & Software*, 20(2), 269-277.
Sonnenberg, J. (1998) New method for the design of the reallocation plan in land consolidation projects. FIG conference.
Thomas, J. (2006) What's on regarding land consolidation in Europe? Proceedings of the XXIII International FIG Congress, Munich, Germany, October 8-13.
Van Dijk, T. (2003) *Dealing with Central European Land Fragmentation*. Eburon, Delft.
Yeh, A. and Qiao, J. (2004) Component based approach in the development of a knowledge based planning support system (KBPSS). Part 1: the architecture of the KBPSS. *Environmental and Planning B: Planning and Design*, 31, 517-537.
Zhu, X. and Healey, R. (1992) Towards intelligent spatial decision support: integrating geographical information systems and expert systems. Proceedings of GIS/LIS '92.

Towards a "typification" of the Pedestrian Surrounding Space: Analysis of the Isovist Using Digital processing Method

Thomas Leduc, Francoise Chaillou, Thomas Ouard

CERMA laboratory UMR CNRS, Nantes, France
{thomas.leduc, francoise.chaillou,thomas.ouard}@cerma.archi.fr

Abstract. The aim of this paper is to couple the isovists field (a useful tool to determine the surroundings) with a classical digital signal processing method so as to classify the open spaces all along a pedestrian pathway and identify some urban patterns. Indeed, it could be of a great interest to determine automatically the type of surrounding spaces to improve the knowledge of the urban fabric at an intermediate level (the one of someone immersed in the city) and to make it possible to enrich its visual perception in real time using dedicated numerical devices. After a brief overview of visibility analysis methods, we focus on the isovist one. The remainder of this paper is dedicated to the methodology of visualscape fingerprint characterization we developed. At last, before concluding, we present a use case based on a real pathway.

1 Introduction

Morello and Ratti (2009) notice that there were "many attempts to translate visual-perception research into architectural and urban design. The best known contribution in urban-planning studies is perhaps Lynch (1960)". In his book, Lynch asserts "We are continuously engaged in the attempt to organize our surroundings, to structure and identify them […] it should be possible to give [cities] a form which facilitates these organizing efforts rather than frustrates them." As explained by Morello and Ratti (2009), city mental maps can help to describe a sort of image of the city but also to

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_14,

evaluate the 'legibility' of a built context. Based on this concept of legibility, Lynch (1960) introduces the derived notion of "imageability" which is a kind of indicator of the evocation power of an environment.

These two key concepts have been enriched by a third one introduced by the Space Syntax theory. Indeed, Hillier (1996) defines the notion of intelligibility as "the degree to which what we can see from the spaces that make up the system – that is how many other spaces are connected to it – is a good guide to what we cannot see, that is the integration of each space into the system as a whole."

Benedikt (2008) reminds us that "Walls and ceilings, buildings and trees, are positioned in such a way as to modulate experience: not just the experience of those very walls and ceilings (and buildings and trees), but the experience of the people and signs, images and machines, and so on, that move about and populate the room or cityscape." To this end, the "theory of isovists" was developed (Benedikt, 1979).

In Franz and Wiener (2008), several isovists measures have been translated into basic spatial qualities hypotheses. Meilinger et al. (2009) address the interactions between partial isovists fields and human wayfinding performance. In Leduc et al. (2009), partial isovists fields have been used to exhibit the fact that it is worth taking strategic visual properties into account in the design of a patrimonial tour in a historic city center. Weitkamp (2010) uses isovist to establish the relationship between landscape openness and wellbeing.

These four last bibliographic references demonstrate, if necessary, the fact that a spatial concept such as the isovist is still relevant today to analyze the plenum conception (Coucleclis 1992) of urban spaces. However, even if this useful tool to model pedestrian perception has already been studied and used, it appears clearly that it has not yet been coupled with a commonly used tool of the signal theory: the frequency analysis. Because perception of visual dynamics involved in motion perspective clearly implies a great set of mental images of the cities, it seems logical to process corresponding digital images of the city as classical digital signals that are often analyzed or modelled in terms of their frequency spectrum.

Such a digital process will help, as an example, to determine the type of surrounding space much more accurately than with traditional isovist's shape indicators. That is to say, it is a helpful tool to improve the knowledge of the urban fabric at an intermediate scale (the one of someone immersed in a city). Such knowledge could, as an example, be useful to enrich the visual perception of a pedestrian in real time (using a dedicated numerical device with augmented reality capability).

The aim of this paper is to present a new relevant isovist's shape indicator. The one presented here is based on the analysis of the complex module of the discrete Fourier transform of the isovist.

2 Overview of isovists’ based methods to analyze open spaces morphology

2.1 Brief overview of visibility analysis

In the 1970s, two main approaches emerge in the visibility analysis context: the concept of viewshed in terrain and landscape analysis and the concept of isovist in architecture and urban space.

The viewshed analysis is a traditional way of analyzing a visibility field. It is defined as the part of the terrain visible from a viewpoint and is basically applied to the landscape with terrain and topographic differentiation (Lynch 1976). As noticed in Yang (2007), viewshed analysis in GIS is rarely applied to urban settings because the operation is based on raster data or TIN (triangular irregular network) data structure, which have problems of accuracy in representing complex geometry of urban form.

An isovist is the set of all points in an environment of opaque surfaces that are visible from a given point (the limit of the isovist is an artificial one functioning something like a horizon in the absence of any other intervening surfaces). This 2D bounded polygon is a useful tool to define the open space concept. From a morphological point of view, open spaces are usually defined as the empty space, the void between the surrounding buildings. However, although these open spaces are not material components of the physical world, they can be conceived as part and parcel of our urban heritage (Teller 2003). Batty (2001) puts the emphasis on the fundamental motivations of conducting visibility analysis research. He noticed that the key questions "how far can we see," "how much can we see," and "how much space is enclosed" are relevant to develop good urban design.

Essentially, isovists describe local geometrical properties of spaces with respect to individual observation points and weigh all the possible view directions equally (see Figure 1). An isovist is a 2D horizontal slice of pedestrian's surrounding space.

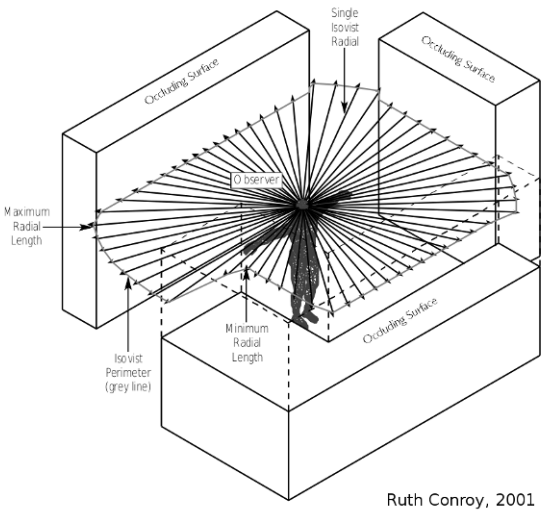


Fig. 1. Isovist symbolic representation (Conroy 2001). The observer is comparable to a visual sensor. Corresponding isovist is the set of all points visible from his given punctual position taking surrounding occluding surfaces into account.

2.2 Analyze the visual dynamics of the pedestrian mobility in the urban fabric: mainly scalar indicators

As written in Davis and Benedikt (1979) and Benedikt (2008), every point in an environment generally has a uniquely shaped isovist belonging to it. Benedikt (2008) defines five useful measures: the area of the isovist (A), the perimeter of the isovist (P), a measure of the length of the radial (Q), a statistical measure of the variability of the boundary's distance from viewpoint (M2), and a measure of the asymmetry of M2 (M3). He noticed that our impression of spaciousness is evoked by high A, low P, low Q, and high M3 (M2 seemed to make little difference). City spaces and parts of them that have these characteristic values - relative to the local norm - will be perceived as more spacious than those with any other combination.

Conroy-Dalton and Dalton (2001) and Weitkamp (2010) calculate some other isovist's geometrical properties such as:

- the area to perimeter ratio,
- the circularity (area of a perfect circle whose radius is set to the mean radial length of the isovist divided by the isovist area),

- the drift (distance between the view-point, i.e. the location from which the isovist is generated and the centre of gravity of the isovist), and
- the minimum, mean, and maximum radial lengths.

The aim here is to characterize the isovist shape using a relevant indicator. The one mentioned before seems to be inaccurate for some different reasons. Perimeter, area, minimum, mean, and maximum radial lengths, but also circularity, are too directly connected with the shape's scale. The required indicator has to be a dimensionless quantity (independent of any linear change of scale). The drift is a measure of displacement between the centroid of the isovist and its viewpoint. Therefore, as the circularity or the area to perimeter ratio, it is a useful tool for measuring the difference between actual and ideal geometric shapes. Such an isovist's surface property does not match our requirement because of the jaggedness of such a shape in the urban context. Moreover the drift parameter is not adapted, because in a given canyon street for each pedestrian's punctual position, the isovist remains unchanged (so as its own centroid) whereas the view-point's position changes.

Finally, the standard deviation of the isovist's radials (M2) and their skewness (M3) measure respectively the "width" of the distribution of radials’ lengths and the "asymmetry" of the distribution of lengths (it indicates if the values are relatively evenly distributed on both sides of the mean radial length).

To sum up, a one-dimensional indicator seems inaccurate to deeply describe a shape as complex as an isovist in an urban outdoor context. Instead of trying to combine some of them so as to produce a composite indicator, we prefer to try a digital signal processing tool.

2.3 Need of a geoprocessing approach - declarative Geoprocessing with the Gearscape Geoprocessing Language

An urban pedestrian pathway and a visualscape are both data that include a spatial component. What is required here is some tool able to process these spatial data using, on one hand, the table-oriented programming paradigm (for its table-friendly syntax, its fundamental and consistent collection operations, and its ease of understanding) and on the other hand, batch processing with parametric ability and procedural extension. We pretend that the use of a spatial SQL with semantics ability is essential to perform such an objective. That is the reason why we need to take benefits from the GearScape Geoprocessing Language (GGL) specific layer (González Cortés and Leduc 2010), aside the features of robustness, scalability, and easy to use main characteristics.

3 Methodology: towards an isovist fingerprint

The aim here is to characterize the isovists' shapes using a relevant indicator. The method we used is based on a "surface ray casting" strategy presented in the 3.1 and 3.2 subsections. The simplified schema presented in Figure 2, sums up the whole spatial process we have developed. It is composed of five main tasks.

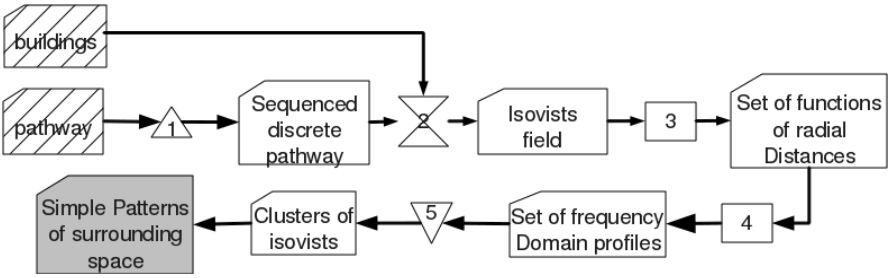


Fig. 2. The processing schema that has been adopted. The sequence is composed of 5 main operations. Input maps are 45° wide hatched. Intermediate results have no background colour and final result is coloured in grey.

The two first tasks are pure spatial processes that strongly involve GGL spatial abilities such as its efficient spatial index implementation. They respectively aim to sample the continuous pathway into a set of equidistant punctual positions (first task) and perform the isovists field computation (second task). The third task consists in an angular abscissa sampling process. The fourth task operates a Discrete Fourier Transform (DFT) of the previous isovists' samples. At last, the fifth task aims to partition the isovists fields using a clustering method so as to classify the pedestrian positions surrounding space.

3.1 Sampling the isovist: 2π periodic function of radial distances

As we need to compute shape indicators from isovist data onto a 1D function, the isovist polygon, which is a complex shape, is transformed into a 1D real-valued function. This reduction of the dimension space is achieved through the discretization of the isovist in polar coordinates with a regular angle (see Figure 3). This 1D function is then plotted to obtain a profile that can be seen as a fingerprint of the corresponding isovist's polygon (see Figure 4).

Fig. 3. Bird's eye view of an isovist in the urban fabric. The pedestrian punctual position is represented by a black focal point (called the viewpoint). Its corresponding isovist is the dark grey polygon. 32 isovist radials are drawn, regularly spaced by an angle of 11.25°. It is a sampling using angular abscissa.

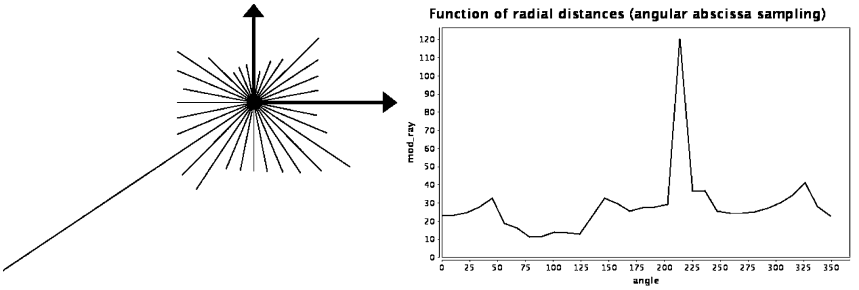


Fig. 4. The previous sampling of the isovist into 32 radials (left hand side) is represented by a profile plot (right hand side). This profile plot corresponds to a function of radial distances all around the viewpoint. In this profile plot, the x-axis corresponds to the angle value in degrees and the y-axis to the corresponding radial length. The angle increases in the counter-clockwise direction.

In practice (in the following use case at least), we sampled the isovist into 1024 radials. It means that the viewpoint’s surrounding space is divided into isovists radials regularly spaced by an angle of about 0.35°. Such an angular abscissa sampling is of enough fine granularity for the urban fabric. Thus, it gives the possibility to detect metric fluctuation at a distance of more than 160 meters. Actually, to be exhaustive, this sampling process must take into account the result exhibited by the Nyquist-Shanon sampling theorem. Thus, the sampling angular abscissa should not be greater

than half the minimum angle between any couple of nodes of the isovist's contour and the view point. As may be noticed, the number of radials is always a power of 2. This constraint is required by the Fast Fourier Transform algorithm we use (Commons-Math 2010).

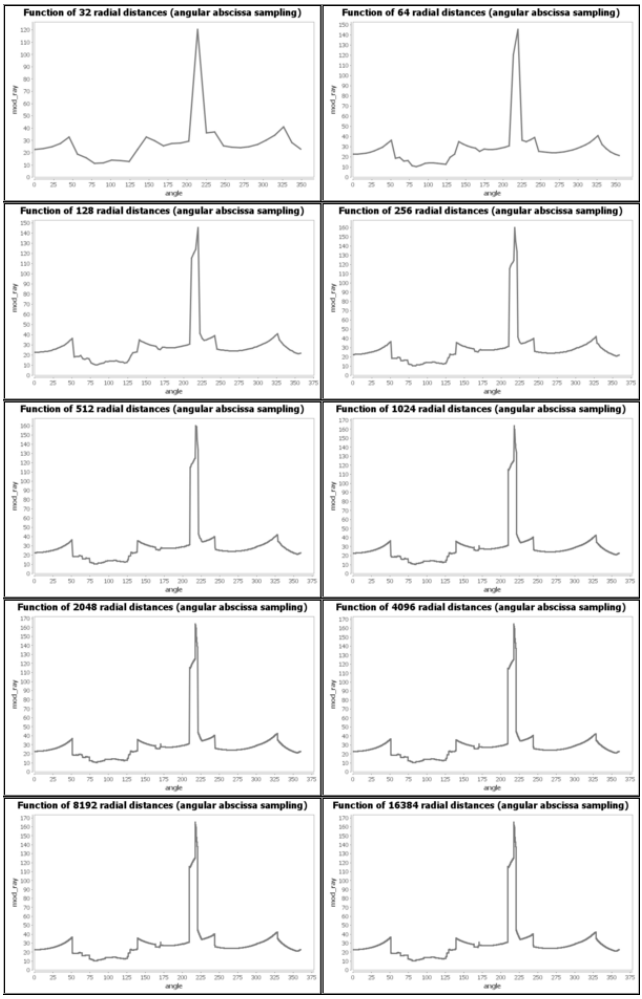


Fig. 5. Sensitivity of the profile plot of the isovist presented in Figure 3 to the sampling frequency. As may be seen on the plots, the process seems to converge from 256 radials.

As shown in Figure 5, in the really specific case of isovist presented in Figure 3, the sensitivity of the profile plot to the sampling frequency seems to converge from 256 radials. With such a simple isovist, sampling into

1024 radials is uselessly accurate (and therefore dispensable), but, generally, such a resolution is well adapted.

3.2 Frequency domain representation of the isovist shape

The continuous Fourier transform is an operation that transforms one complex-valued function of a real variable into another. It consists in decomposition on a sinusoidal basis function of the input function. Because the function of radial distances we use is a discrete one (the continuous border of the isovist has been sampled in the previous subsection), we need a discrete implementation of this transform (DFT stands for Discrete Fourier Transform). For efficiency reasons, we have decided to reuse an already implemented Fast Fourier Transform (FFT) algorithm (see Commons-Math 2010).

Main properties of the FFT are: completeness (it is an invertible and linear transform), periodicity, and shift capability. All these properties seem essential to describe almost equivalent shapes except that some of them have been transformed by a rotation, a non-rotating dilation (a homothecy), etc. Figure 6 corresponds to the Fast Fourier Transform of the function of radial distance presented in Figure 4. As may be noticed, the right hand side of the figure corresponds to a shift of $-\pi$ radians of this 2π periodic function. This second plot exhibits the fact that the DFT complex module has the y-axis as an axis of symmetry.

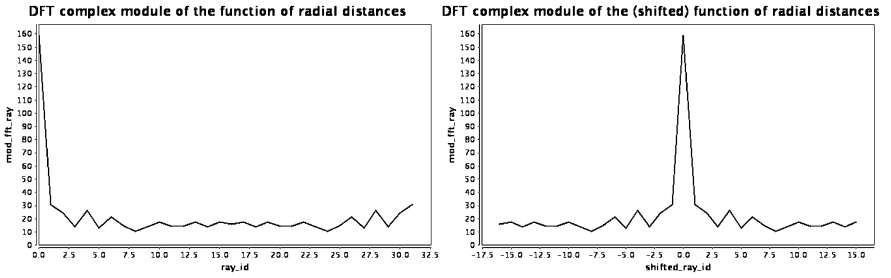


Fig. 6. Profile plot of the complex module of the Fast Fourier Transform of the function of radial distances given in Figure 4. As may be see on the shifted plot (right hand side), it has the y-axis as an axis of symmetry.

Concerning the central peak of the Figure 6, it is due to the fact that the input function (the function of 32 radial distances) is almost constant except on a single value (south west oriented). It is a sort of a "unit impulse function" (one could say a Dirac distribution added to a constant distribu-

tion). The other smaller peaks correspond to weaker fluctuations all around the mean value.

4 Use case: analysis of a real urban area in Nantes city

4.1 An already tried and tested playground

The study site is a pedestrian axis in the historical city centre of Nantes, a west-coast located city in France. It is of about 500 meters in length, starting from *Place du Pilori* square (a medieval district) and ending at *Place Royale* square (with 19th century morphology). Between these two ends, a main street called *Cours des 50 otages* (contemporary style) split the area in two distinct parts (see Figure 7). The choice of this site is not only motivated by its variety and its straightness, but also due to the fact that it has already been studied several times.

As demonstrated by both Tahrani et al. (2006) and Sarradin et al. (2007), the visualscape all along this path is a rhythmic sequence of visual opening and closing imprints.

Fig. 7. A part of the town center of Nantes city with the route followed by the observer. This pathway is sampled into 20 positions. The yellow polygonal area corresponds to the isovists field build on these 20 positions.

4.2 A set of punctual located snapshots

To start with, we have decided to test the method with a small size of sampling positions. The 3-dimensional profile plot presented in Figure 8 (thanks to the GNU-Octave software) aims to reproduce the visual dynamics all along the 20 discrete positions. Its first axis (labelled with a range from 0 to 20) corresponds to the set of pedestrian positions. Its second axis (labelled with a range from 0 to 1024) corresponds to the angular abscissa sampling. Finally, its third axis (labelled with a range from 0 to 450) corresponds to the radials lengths. Even if it is quite complex to understand, several bumps appear clearly. The two biggest and almost symmetrical ones at angular positions 60 and 600 correspond respectively to azimuths of 20° and 200° - that is to the axis itself in both directions. The first one (60^{th} radial, 20°) corresponds to the road towards the *Place du Pilori* square whilst the second one (600^{th} radial, 200°) corresponds to the road towards the *Place Royale*.

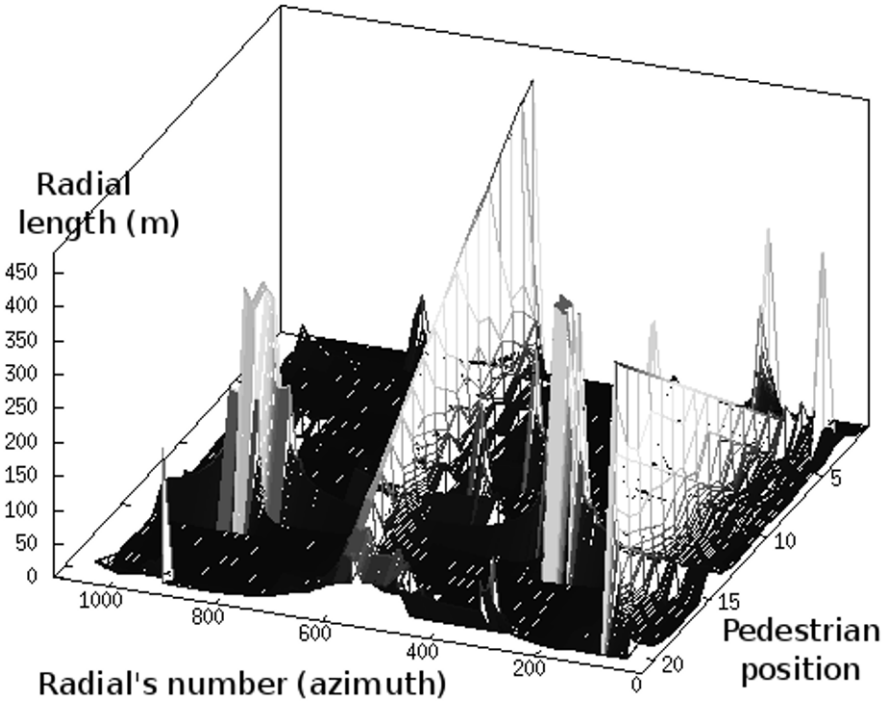


Fig. 8. A 3D profile plot of the isovists field along the 20 positions of the pedestrian.

The two peaks at positions 13 and 14 correspond to the *Cours de 50 otages* main street in the same way. To simplify the understanding of this profile plot, we have unfolded it into 20 2-dimensional profile plots (see Figure 9).

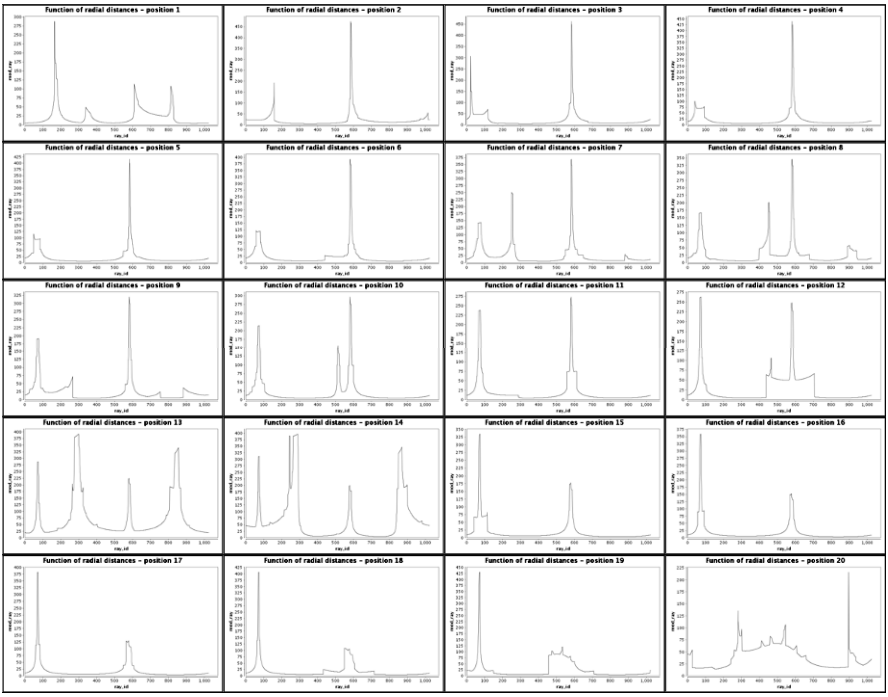


Fig. 9. Set of 20 profile plots (one per pedestrian position). The quite ever-present peaks around positions 60 (20°, azimuth of the road towards the *Place du Pilori* square) and 600 (200°, azimuth of the road towards the *Place Royale* square) correspond respectively to the vantage of the road towards the *Place du Pilori* square and to the vantage of the road towards the *Place Royale* square.

As described in our methodology, the next step is to transform all these functions of radial distanc`es into the frequency domain. As shown in Figure 10, the complex modules of the FFT appear to share some similarities. The aim of the next section is to establish them a bit more precisely.

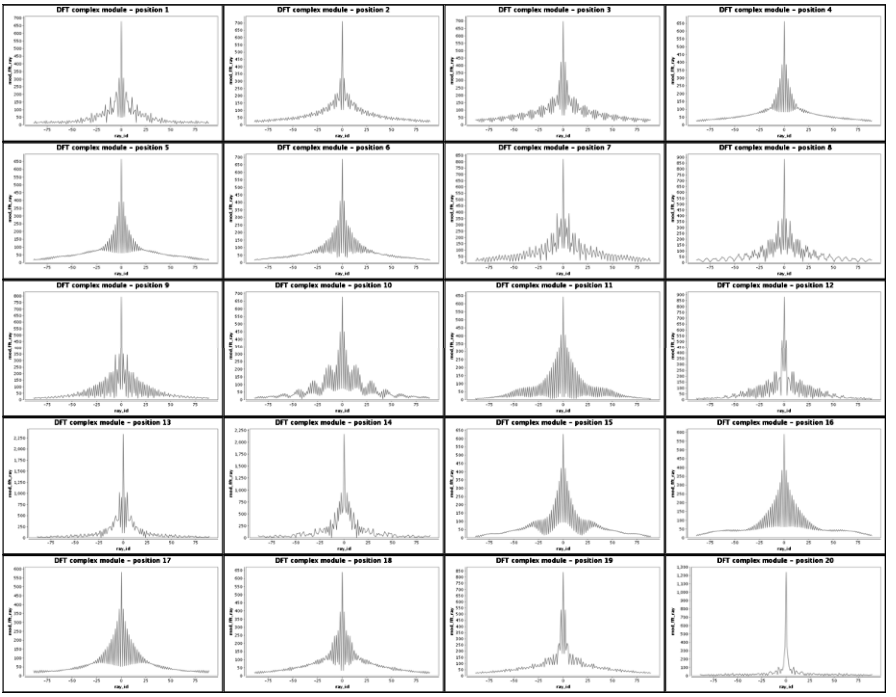


Fig. 10. Profile plots of the complex module of the Fast Fourier Transform of the functions of radial distances for each pedestrian position.

4.3 Classification

The aim here is to partition the 20 observations (the 20 sets of 1024 values produced by the FFT) into a few sets of clusters. Because of the small size of the sampling, we have chosen a hierarchical clustering method. The dendogram presented in Figure 11, shows nearness between the 1024 coefficients of the FFT of the 5, 6, 9, 10, 11, 15, 16, and 17 positions (canyon streets with far vantages). This cluster is clearly far, on the one hand, from the one that encloses positions 13 and 14 (crossroad with a main street), and on the other hand, from the one of the position 20 (square entrance). At last, a fourth cluster appears merging all other types of road junctions.

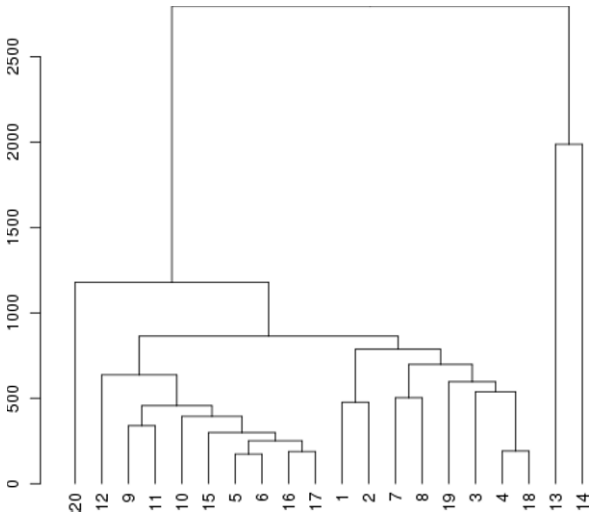


Fig. 11. Result of the hierarchical clustering method.

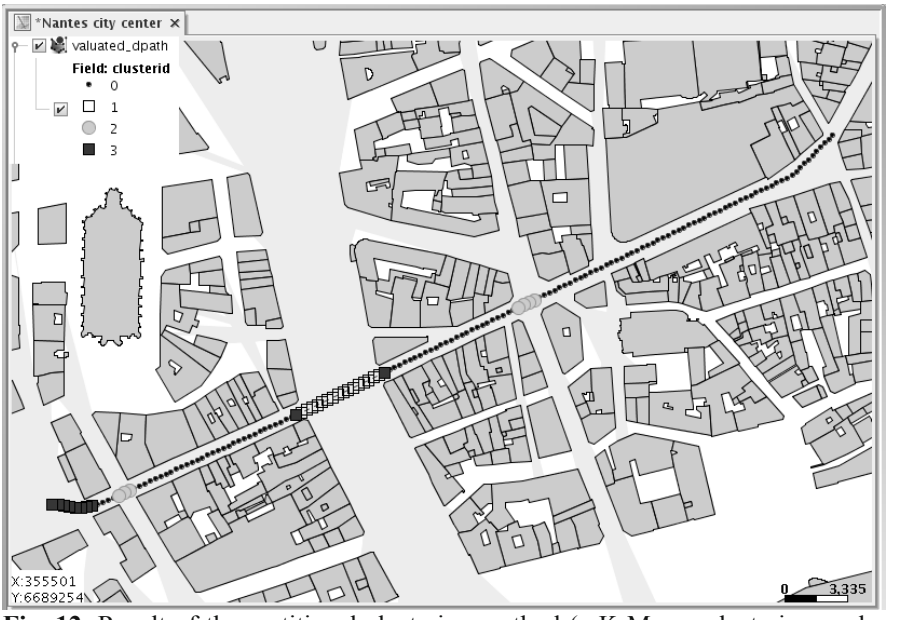


Fig. 12. Result of the partitional clustering method (a K-Means clustering analysis).

To test the scalability of this indicator, we decide to focus on a greater sample (153 positions). Because of this sampling size, we adopt another classification method based on partitional clustering. The implementation

we use is the K-Means of Math-Commons (2010). Re-using the intermediate result produced by the hierarchical clustering, we have arbitrarily decided to choose 4 as the number of clusters. What clearly appears in Figure 12 is that this new result is not only fully coherent with the previous one, but also that the results seem qualitatively true and spatially accurate.

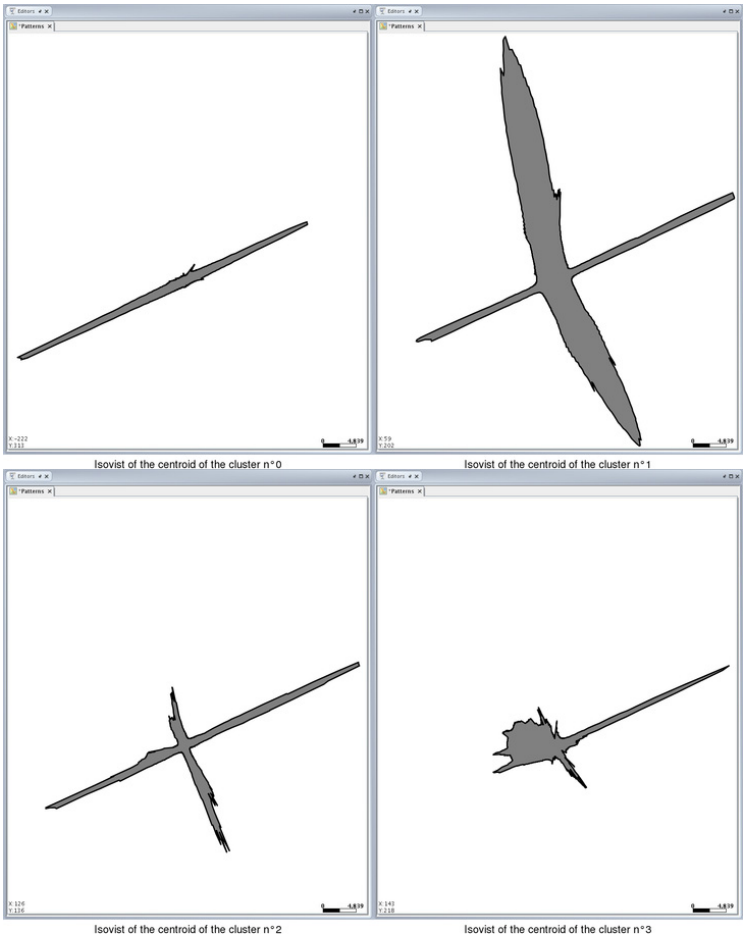


Fig. 13. Sketch of an emblematic shape per cluster. Each one corresponds to the values given by the centroid of the corresponding partition.

4.4 The identified patterns

As mentioned in the previous section, four clusters have been identified using two different classification methods. To go a bit further, we have decided to plot the most emblematic isovist shape for each cluster. The results presented in Figure 13 correspond to the centroids of each partition.

One can notice that the first shape (upper left) matches to a straight line canyon street. The second one (upper right) corresponds to a crossroad in between a main street and a smaller street. The third one (lower right) approximates a T-junction, while the fourth one matches a square entrance.

5 Concluding remarks and outlook

This paper presents a new method to classify the pedestrian surrounding space based both on isovists field and digital signal processing. This shape indicator has been developed in the context of the GGL geoprocessing language so as to benefit from efficient space queries implementations. The first results obtained on the Nantes city centre are a green-light that illustrate its potentialities for visuals dynamics' evaluation all along pedestrian pathways.

Nevertheless, one must admit that the experimental site is rectilinear. With a (nore) sinuous tour, the complex module of the FFT will probably not be enough of a differential tool. A solution could be to couple the complex module analysis with the complex argument or phases of the FFT.

Concerning our future works, because we really think that this level (we mean the surroundings of a pedestrian) is the right one to precisely model the city, we would like to extend this approach to the whole city. Thus, a fine and quite precise model of all open spaces would be available, so as a classification of each of them. Such a classification is a first step towards a compressed (and therefore possibly embedded on mobile devices) model of the city.

References

Batty, M. (2001). Exploring isovist fields: space and shape in architectural and urban morphology. Planning and design: Environment and planning B, 28(1):123-150.

Benedikt, M. L. (1979). To take hold of space: isovists and isovist fields. Environment and Planning B: Planning and Design, 6(1):47-65.

Benedikt, M. L. (2008). Cityspace, cyberspace and the spatiology of information. Journal of Virtual Worlds Research, 1(1):22.

Commons-Math (2010). The Apache Commons Mathematics Library. http://commons.apache.org/math/.

Conroy, R. (2001). Spatial Navigation in Immersive Virtual Environments. PhD thesis, The faculty of the built environment, University College London, London, U.K.

Conroy Dalton, R. and Dalton, N. (2001). OmniVista: an application for Isovist field and path analysis. In 3rd International Space Syntax Symposium, Atlanta, Georgia, USA.

Couclelis, H. (1992). People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS. In Frank, A. U., Campari, I., and Formentini, U., editors, Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning, Lecture Notes in Computer Science, pages 65-77, Pisa, Italy. Springer.

Davis, L. S. and Benedikt, M. L. (1979). Computational models of space: Isovists and isovist fields. Computer Graphics and Image Processing, 11:49-72.

Franz, G. and Wiener, J. M. (2008). From space syntax to space semantics: a behaviorally and perceptually oriented methodology for the efficient description of the geometry and topology of environments. Environment and Planning B: Planning and Design, 35(4):574-2013592.

Gibson, J. J. (1979). The Ecological Approach to Visual Perception. Boston: Houghton Mifflin.

González Cortés, F. and Leduc, T. (2010). Poster abstract - GGL: A geoprocessing definition language that enhance spatial SQL with parameterization. In 13th AGILE International Conference on Geographic Information Science – AGILE’2010, Guimaraes, Portugal.

Hillier, B. (1996). Space is the machine. Press Syndicate of the University of Cambridge.

Leduc, T., Miguet, F., Tourre, V., and Woloszyn, P. (2010). Towards a spatial semantics to analyze the visual dynamics of the pedestrian mobility in the urban fabric. In Painho, M., Santos, M. Y., and Pundt, H., editors, Geospatial Thinking (associated to the 13th AGILE International Conference on Geographic Information Science, Guimaraes, Portugal - AGILE'2010), Lecture notes in Geoinformation and Cartography (LNG&C), pages 237-257. Springer-Verlag, Berlin Heidelberg.

Lynch, K. A. (1960). The image of the city. Cambridge: MIT Press. Publication of the Joint Center for Urban Studies.

Meilinger, T., Franz, G., and Bülthoff, H. H. (2009). From isovists via mental representations to behaviour: first steps toward closing the causal chain. Environment and Planning B: Planning and Design. Advance online publication.

Morello, E. and Ratti, C. (2009). A digital image of the city: 3D isovists in Lynch's urban analysis. Environment and Planning B: Planning and Design, 36(5):837-853.

Sarradin, F., Siret, D., Couprie, M., and Teller, J. (2007). Comparing sky shape skeletons for the analysis of visual dynamics along routes. Planning and design: Environment and planning B, 34(5):840-857.

Tahrani, S., Moreau, G., and Woloszyn, P. (2006). Analyzing urban daylighting ambiences by walking in a virtual city. In Premières journées de l'AFRV, Rocquencourt, France.

Tahrani, S. and Moreau, G. (2008). Integration of immersive walking to analyse urban daylighting ambiences. Journal of Urban Design, 13(1):99-123.

Teller, J. (2003). A spherical metric for the field-oriented analysis of complex urban open spaces. Planning and design: Environment and planning B, 30(3):339-356.

Yang PPJ, Putra SY, and Li W (2007). Viewsphere: a GIS-based 3D visibility analysis for urban design evaluation. Planning and design: Environment and planning B, 34(6):971-992.

Weitkamp, G. (2010). Capturing the view: a GIS based procedure to assess perceived landscape openness. PhD thesis, Wageningen University, The Netherlands.

Modelling Umwelten

Jens Ortmann, Henry Michels

Institute for Geoinformatics, University of Muenster, Muenster, Germany
{jens.ortmann, henry.michels}@uni-muenster.de

Abstract. In this paper, we present a theory to structure spaces of human activities and to connect these spaces with geographic information. We introduce a notion of environment originating in the work of von Uexküll termed Umwelt. The Umwelt links agents and the observations they make with the activities they perform using the objects in their environment that afford these activities. We then provide a mereology of environments determined by activities. We use a scenario of travelling to work to illustrate at hand of a practical application how we can integrate sensor observations in our theory to support decision-making in every-day activities. As a test case, we deal with the event of icy roads derived from current weather conditions. Our theory is implemented as ontology in the Web Service Modelling Language (WSML). To demonstrate our ontology in practice, we simulate sensor observations and their impact on the ontology in an Agent Based Simulation.

1 Introduction

Nowadays, environmental data is mostly acquired through sensors or whole sensor networks. The technical progress made sensors cheaper and easier to use. This resulted in a growing number of sensors. The technical progress was paralleled by developments on sensor description, sensor access and data processing. The Open Geospatial Consortium[1] (OGC) advances developments to make observations made by sensor networks as well as archived sensor data easily accessible on the web. These observa-

[1] for more information visit http://www.opengeospatial.org (last accessed 15.12.2010)

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_15,

tions can for example be discovered, accessed, and controlled using open standard protocols and interfaces provided by the Sensor Web Enablement (SWE) initiative [3]. The sheer amount of available data is difficult to handle for non-experts. Most users are only interested in specific information for their purpose.

The following scenario introduces our vision of a small notification system, that we realize using the ontology that we present in this paper.

John lives in Bunnik, about seven kilometres from Utrecht where he works. Usually he takes the bike to work. John would like to know when it is impossible to cycle to work, so that he can schedule his day, cancel meetings at the office, and inform his boss. He found a web portal providing notifications about events affecting certain activities. Hence, John is specifying his activity "driving-to-work-by-bike," a spatial extent for it, and chooses an SMS-based notification. The designers of the portal describe natural events in the environment using real time values of several sensor networks. Once an event affecting John's activity, like the "icy-road"-event, is triggered (temperature of the ground is lower than 0◦C and the amount of precipitation is greater than 0mm/h), John gets a text message that he will not be able to cycle to work today.[2]

Our approach to allow more precise information retrieval is centred on specifying the activities of a user. We introduce von Uexküll's notion of *Umwelt* [33] to refer to human environments determined by agents and their activities. We provide a theory implemented as ontology that is based on work by von Uexküll, but also draws from work by Barker [1] and Gibson [8, 9] as well as more recent work by Smith and Varzi [28, 29]. The theory gives an account of the environment that is agent-specific and considers activities and objects as central elements. Following [14], we avoid philosophical discussions and focus on practical applications based on ecological and psychological theories. Therefore, the resulting ontology is particularly useful in dynamic scenarios and in scenarios where the function and use of objects are given priority over their individual properties.

We demonstrate the practicability and value of our theory by running a simulation based on our ontology in Agent Based Simulation software. The scope of this paper can be summarized as follows:

- A (formal) theory of human environments that
  - connects humans, their activities, the objects that are required for these activities, and the spaces in which humans are active;
  - structures spaces of human activities mereologically;
  - bridges between foundational models and practical applications; and

[2] The scenario makes abstracting assumption and it would be possible to model freezing events more accurately with our approach. However, for reasons of simplicity and clarity we only consider precipitation and temperature as crucial factors for icy roads.

  - is implemented as ontology.
- A prototypical application resting on this formal theory that
  - uses and extends the Umwelt theory;
  - implements the notification scenario outlined in this Section; and
  - is demonstrated in an agent-based simulation.

The next sections will explain the theories behind the approach (Section 2), the conceptualization of our theory (Section 3) and its implementation (Section 4), as well as a small application to prove the feasibility of our approach (Section 5). After that, we give an outlook on future work (Section 6) and a conclusion (Section 7).

2 Background and Related Work

The two fundamental things agents do in their environment are *perceiving* and *(inter)acting*. In this paper, we adopt a very basic notion of agent from artificial intelligence:

An **agent** is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon its environment through **effectors** [23, p. 31].

This definition comprises at first all animals, including humans, which makes this notion compatible with theories from ecology and psychology. Furthermore, all robotic and other agents used in artificial intelligence fit under this description. Finally, also groups, e.g., organizations and institutions are agents, may be included as long as they are regarded as perceiving and acting as individuals.

Notably, this definition of an agent can already be found in von Uexküll's work on the environment of animals and humans [33]. Von Uexküll's *Funktionskreis* (functional circle) characterizes an animal as something that perceives its environment through senses and acts upon it through effectors (c.f. Figure 1). The Funktionskreis is also a semantic model of environmental perception. It describes our encounter with the environment as a loop of perception and action where the possibilities for action override the sensual perception and henceforth determine the meaning of the objects in the environment. The Funktionskreis forms the basis of von Uexküll's environment theory (orig. "Umweltlehre"). The environment of an animal is given by the functional meanings of the objects in that environment, i.e. by the opportunities for action.

The idea of opportunities for action was later investigated prominently by Gibson [8, 9]. Gibson's *affordances* are opportunities provided to an animal by the environment. Some scholars complement affordances as

properties of things in the environment with properties (capabilities) of the agent [25, 32], which then fits nicely von Uexküll's Funktionskreis.

Gibson provided his own theory of environments based on his notion of affordances. However, the concept of affordance has not reached a mature state and is hardly found in formal ontology. Furthermore, Gibson assumed an environment shared by agents with similar capabilities (i.e. animals of the same species), whereas von Uexküll's Umwelt is an environment for individuals. Unfortunately, neither von Uexküll nor Gibson provided formal theories and the formalisations of Gibson's theories proposed by later scholars (e.g. [32, 31, 4]) have never reached a state of common agreement.

In formal ontology, a mereotopological first-order theory of the *Niche* was proposed by Smith and Varzi [28, 29]. The niche theory accounts for the relations between an agent and its environment. The formal theory focuses on mereological (i.e. part of) and topological (i.e. boundary for) relations. Smith and Varzi's informal theory rests on Gibson's [9] environment, von Uexküll's [33] Umwelt, and Barker's [1] physical-behavioural units. The latter are units of the agent-environment system given by a physical object and the behaviour that it affords to the agent.

Recently, Bennett presented some fundamentals on ontology of environment and habitat [2]. His goals are disambiguating different uses of the term "environment" and identifying terms that should be covered in an ontology of environment and habitat. Even though, lining out notions of space, time, and geographic regions formally, he does not employ this formal theory when characterizing environments. Bennett distinguishes between an immediate and an effective environment on a level of functional interaction and between a global and a local environment on a level of spatial extent. The immediate environment comprises the direct impacts of the environment on the agent's outer surface, i.e. its boundary to the environment. The effective environment is composed of the objects that are related to the agent. Unfortunately, Bennett does not provide any references to support his distinction. Von Uexküll's Funktionskreis actually shows how Bennett's immediate environment collapses under his effective environment, rendering the distinction unnecessary. The local environment forms the proximity of the agent, whereas the global environment is basically the sum of all possible local environments. Again, Bennett's distinction is not grounded in any theory nor does it point to any reference. Suggestions for a distinction according to the spatial extent of environments have been made by von Uexküll [33, p. 42ff.] based on human capabilities for 3D vision and also Granö [11, p. 18ff. and p. 108ff.] provides differentiation between the "proximity" and the "landscape" based on visual abilities to judge sizes, distances, and plasticity of objects. More recently, Mon-

tello [18] discussed different classifications of environmental scales and suggested a distinction based on spatial apprehension.

3 A Theory of Nested Umwelten

In this section, we provide a theory of Umwelten, i.e. von Uexküll's notion of agent's environments [33]. We use the original German term *Umwelt*[3] (pl. Umwelten) to refer to von Uexküll's environment of semiotic action-perception cycles. The term *environment* is used ambiguously in various disciplines, whereas the term *Umwelt* is a technical term in semiotics [6]. Our theory provides a mereological structure of nested Umwelten, partially applying the formal theory of Smith and Varzi [28]. It also borrows from work of Barker [1], Gibson [8, 9], and further work of Smith and Varzi [29].

3.1 The Umwelt

The original sense of an agent's Umwelt coined by von Uexküll is described as a small excerpt of physical reality that is tailored according to an agent's capabilities for interaction in the environment [33, p. 101]. Crucial to an understanding of von Uexküll's Umwelt notion is his *Funktionskreis* that describes how our receptors and our effectors *embrace* an object in the Umwelt. The Funktionskreis is depicted in Figure 1. An agent, equipped with sensors and effectors, interacts with the object of his environment perceiving the sense sign with its receptors but also perceiving the effect sign according to its effectors. As the loop between sensor and effector closes on the object side (through the so-called opposite structure) the effect sign overrides the sense sign, resulting in the perception of the object's opportunities for action for the agent. The Funktionskreis summarizes von Uexküll's semiotic theory of functional meanings of objects in the Umwelt. According to this theory, von Uexküll's Umwelt is a construct of perceived objects. The Umwelt consists of the objects that have functional meaning for an agent. Therefore, the objects of the Umwelt are not objective physical objects but *painted* according to subjective functional significances for an agent [33]. Hence, the Umwelt is foremost a functional environment. We abstract from temporal constraints here. Dealing with problems like a street affording safe travel during the day but not at night has to be postponed to future work.

[3] The German term "Umwelt" can be translated as "surrounding world", or "environment" in English.

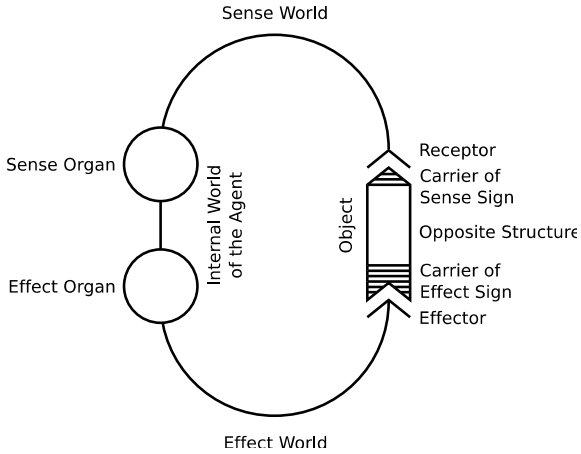


Fig. 1. Von Uexküll's Funktionskreis as depicted in [33, p. 27], our translation. Note: We prefer the notion of agent to the term subject (German: Subjekt) used by von Uexküll.

We adopt Smith and Varzi's [28, 29] mereological nesting of environments. Yet, we do not provide an account of topology and location. We see the functional and the spatial account as perpendicular dimensions of environments and would even add a temporal dimension. However, its treatment eludes the scope of this paper. Instead, our Umwelt theory makes activities and events explicit. In doing so, we aim at a functional and dynamic account of environments. This is important in practical applications that simulate changes of the environment and their impacts on agents and their activities.

The notion of *fitness* is another important idea in ecological theories and also appears in the niche theory by Smith and Varzi. It describes how an agent perfectly fits into its environment. That is, the skin or shell of the agent forms a seamless boundary between agent and environment. Looking at this more functionally, instead of spatially, the boundary is determined by the sensors and effectors (which for many animals, results in the same boundary, as the skin is equipped with receptors). Furthermore, as Gibson stated:

> No animal could exist without an environment surrounding it. Equally, although not so obvious, an environment implies an animal (at least an organism) to be surrounded. [9, p. 8].

This is particularly true for the Umwelt defined by functional significances and we can simply extend this claim to all agents, not only animals. It is

obvious that without an agent with effectors the Umwelt cannot exist, and likewise, every agent's surroundings form an Umwelt affording interactions. This reciprocal relation is depicted in Figure 2.

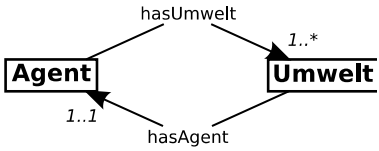


Fig. 2. The two concepts `Agent` and `Umwelt` imply each other, every agent has an Umwelt it lives in and every Umwelt has an agent that it surrounds. An agent can also have more than one Umwelt, whereas an Umwelt is specific to exactly one agent.

Von Uexküll's Umwelt is made up of objects. We borrow this view and extend it in that objects are characterized through states. The view that objects constitute the environment and that we can perceive objects is not shared by all scholars. However, we think that the general theory of Umwelten we present here can be easily adapted to other theories of perception, as long as they allow perception of and interaction with their basic primitives. Additionally, using objects as primitives of the Umwelt simply has practical advantages in our application (cf. Section 5) and also has the advantage to be intuitive to non-experts in psychology and philosophy of perception. Parallel to the reciprocal relation between agent and Umwelt, there are two types of events that can occur in the Umwelt. We distinguish between events caused by the agent and events caused by the Umwelt. Events caused by agents are their *activities*.

Activities are tightly linked to the Umwelt. They are determined by the objects and the opportunities these objects afford to the agent. We use activities to define the boundaries of specific Umwelten by constructing Activity Umwelten (see Section 3.3). In line with Galton [7], we define an event as something happening at a certain place and time. Additionally, Galton describes the impact of events on objects as the change of states of these objects. However, we only deal with natural events and term them simply *events*.

We can sum up the basic characteristics of the Umwelt in our theory as follows:

(1) the Umwelt is a functional environment for an agent;
(2) the agent fits perfectly into its Umwelt;
(3) Umwelt and agent imply each other, that is every Umwelt has exactly one agent and every agent has at minimum one Umwelt;

(4) the Umwelt is made up of objects that afford actions to the agent; and
(5) the objects are affected by events of the Umwelt and by activities of the agent in the Umwelt.

In the following, we will introduce three more specific notions of Umwelt, namely the Universal Umwelt, the Activity Umwelt, and the Compound Activity Umwelt.

3.2 The Universal Umwelt

Providing a mereological structure of different Umwelten we first introduce the *Universal Umwelt*. The Universal Umwelt is the most general Umwelt of an Agent, it contains all objects that the agent can interact with, and has all the affordances that exist for the agent. The Universal Umwelt is defined as the mereological sum of all other Umwelten (mostly Activity Umwelten described in Section 3.3) that an agent has. Therefore, an agent has only one unique Universal Umwelt.

(6) The Universal Umwelt is defined as the mereological sum of all other Umwelten of an agent.
(7) An agent has exactly one Universal Umwelt.

The Universal Umwelt is mostly useful as an umbrella concept to group all other Umwelten. It allows formulating constraints that apply to all its parts, i.e. all the Umwelten of an agent.

3.3 The Activity Umwelt

The focus in our work is on what we call the *Activity Umwelt*. An Activity Umwelt is the Umwelt of an agent for one specific activity. It comprises all the objects that are required for this activity and takes up the space that affords the activity. As an environmental unit, it comes close to Barker's physical-behavioural units [1]. In particular, we are interested in what Barker calls modal behaviour, which are actions on the level of agents (not on the level of organs). Hence, the Activity Umwelt of, for example, Laura's activity of making her coffee is defined by her coffee-machine, the box where she keeps her coffee, and the shelf where she keeps that box, as well as the spoon and the coffee filter, and the part of her kitchen that she needs to perform this activity. There is a perfect match between Laura's

Activity Umwelt and her, due to the physical presence of all the things that she needs to make coffee matching her coffee-making abilities.

Activity Umwelten form a nested hierarchy based on a taxonomy of activities. As illustrated in Figure 3, the Activity Umwelt for cycling is nested within the Activity Umwelt for the more general activity of driving, which in turn is nested within the Activity Umwelt for the even more general activity of locomotion. Because of nesting, an Agent can have several Activity Umwelten at the same time, however, it is not necessary that the different Activity Umwelten an agent has are structured mereologically; they can also be the Activity Umwelten for different but temporally overlapping Activities.

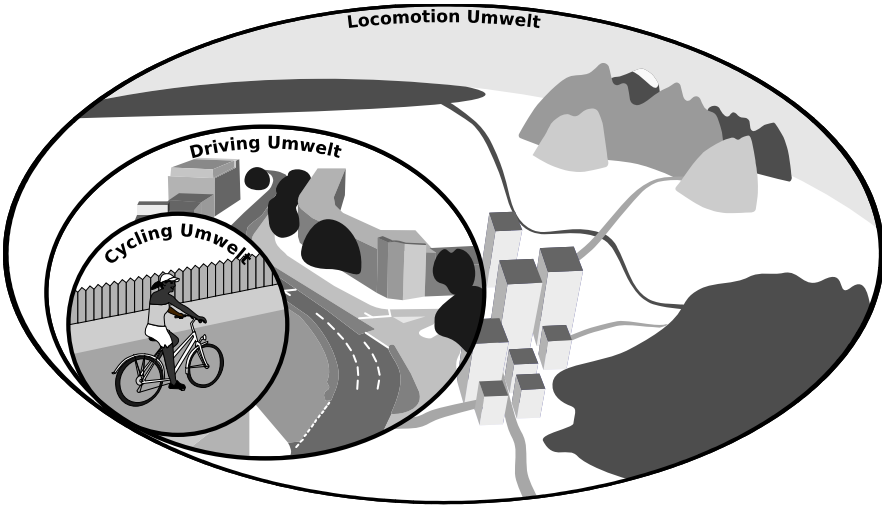


Fig. 3. Illustration of nested Activity Umwelten for an agent. An Activity Umwelt for cycling is nested within an Activity Umwelt for driving, which is nested in an Activity Umwelt for locomotion.

The Activity Umwelt provides us with a tool to bridge between agents, activities, and the space in which these activities take place. All three entities require each other in applications of our theory.

Furthermore, events occurring in the Umwelt can affect the objects due to the change of their states and therefore change the Umwelt. These couplings finally allow us to study the impacts of environmental events on agents and their activities. An illustration of the relations between agent, activity and Activity Umwelt is shown in Figure 4.

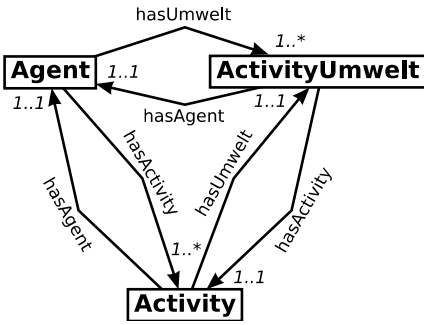


Fig. 4. We can extend Figure 2 with an activity, keeping the agent, and using the more specialized Activity Umwelt. An activity is specific to exactly one agent and gives rise to exactly one Activity Umwelt. An agent can have several Activity Umwelten, one for each activity it performs.

The spatial boundary of an Activity Umwelt is derived from the objects[4] that are involved in the activity. An Activity Umwelt has also temporal boundaries that are given by the temporal boundary of the activity accounted for. The Activity Umwelt for John's cycling-to-work activity has spatio-temporal boundaries. It only exists during the time that he cycles to work and extends as a kind of tube above the road that he takes.

(8) The Activity Umwelt is a physical-behavioural unit;
(9) an Activity Umwelt is the Umwelt for exactly one activity of an agent;
(10) Activity Umwelten can be structured in a mereological hierarchy; and
(11) the Activity Umwelt is part of the Universal Umwelt of the agent.

An activity that gives rise to an Activity Umwelt is a unique and atomic entity in our theory. For the uniqueness, we suggest two criteria of identity for an activity. An activity can be unambiguously identified by the agent that performs it and the time at which it is performed. As additional identity criteria, we can add the objects involved, but agent and time are sufficient in all but borderline cases. If an agent carries two plates at once from the kitchen to the living room, we do not consider this as two activities of carrying one plate, but as one activity of carrying two plates. The atomicity of activities is a claim that we make: Activities cannot have parts.

[4] Following Gibson's ontology [9] the medium (e.g., air) between the objects is also part of the environment; we make this assumption only implicitly.

3.4 The Compound Activity Umwelt

Even though activities do not have parts, they can themselves be part of compound activities. For example, assembling a bookshelf from the do-it-yourself shop can be regarded as a compound activity. We might model the activities according to the instructions provided, one activity for adding some screws, a second activity for connecting the sides to the bottom part, a third activity to fix the top part, and so on. These compound activities give rise to Compound Activity Umwelten. The Compound Activity Umwelt is the mereological sum of one or more Activity Umwelten. This allows defining more complex environments comprising several activities. The decision what to model as compound activity and what to model as activity rests on the user who has to decide which granularity is required for his application.

(12) A Compound Activity Umwelt is the mereological sum of one or more Activity Umwelten.

In Figure 5, we illustrate the concepts and relations used in our theory

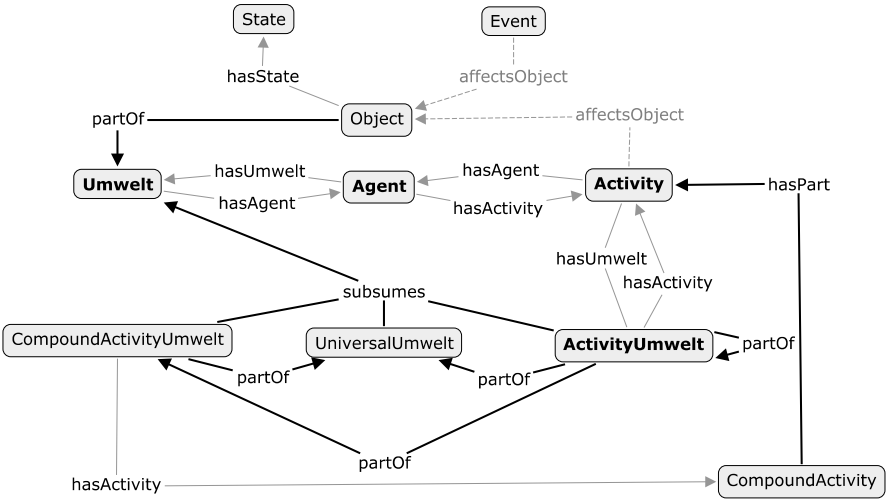


Fig. 5. Concept map of the Umwelt theory. The core concepts are highlighted in boldface fonts. The links for parthood and subsumption relations are made thicker to stress them in the illustration.

4 Formalization in WSML

We formalized our theory as ontology in the Web Service Modelling Language (WSML) [30]. WSML has been submitted to the World Wide Web Consortium[5] for discussion as standard [5]. Implementing our theory in a formal language helps making the theory explicit and checking its logical consistency. Additionally, an ontology encoded in a formal language can be easily shared on the Web and therefore used by other people. Furthermore, WSML has the advantage over other ontology languages (e.g. the Web Ontology Language OWL[6]) that it has a built-in rule support. The rule support is required in many applications that we think the presented ontology is useful for. In particular, our scenario requires processing inputs from sensors and outputting to notification services.

WSML comes in five dialects with varying expressiveness. For our application, we chose the variant WSML-flight. WSML-flight is an extension of the core WSML dialect towards logical programming and adds rule support. WSML-flight allows the specification of *concepts* with *attributes*, global *relations*, *instances* of concepts and relations, as well as *axioms* to constrain attributes and relations, and to specify additional rules. In the remainder we will use true type fonts to indicate concepts like `Umwelt` of our ontology representing real world entities like the Umwelt. We will furthermore refer to our formal theory as Umwelt ontology. The Umwelt Ontology is available online[7].

Our Umwelt ontology has eleven concepts, of which the concept `Entity` is only a placeholder for the most general concept, which has no constraining attributes. At its core, our theory is a theory of agents in their Umwelten. This is represented by two concepts `Agent` and `Umwelt`, which have attributes that express that every Umwelt has an agent and every agent has an Umwelt. We have ensured this using cardinality constraints setting the minimum cardinality of the attributes `hasAgent` of `Umwelt` and `hasUmwelt` of `Agent` to 1. The WSML code shown in Listing 1 expresses Gibson's claim of reciprocal existential dependence between agent and Umwelt (c.f. Section 3).

[5] see http://www.w3.org for more information (last accessed 15.12.2010)

[6] see http://www.w3.org/TR/owl-ref/ for the language reference (last accessed 15.12.2010)

[7] Visit http://trac6.assembla.com/soray/wiki/SwoyAgile to find all ontologies used throughout this paper or http://svn6.assembla.com/svn/soray/user/Henry/publications/AGILE%202010/ontologies/Umwelt.wsml to directly access the Umwelt Ontology as WSML file (both accessed 22.12.2010).

Listing 1 The concepts `Umwelt` and `Agent` with cardinality-constrained attributes connecting each other in WSML. The attributes are written indented below the concept definition. Minimum and maximum cardinality constraints are defined in brackets before the range specification of the attribute.

```
concept Umwelt subConceptOf Entity
    hasAgent ofType (1 *) Agent

concept Agent subConceptOf Entity
    hasUmwelt ofType (1 *) Umwelt
```

The concept `Activitiy` represents an agent's activity in its Umwelt. An `Activity` is linked to exactly one `Agent`, whereas our ontology allows modelling any number of activities for an agent. The activity of an agent directly implies an Activity Umwelt. The `ActivityUmwelt` in our theory is linked to one `Activity` and one `Activity` links to one `ActivityUmwelt`. The `ActivityUmwelt` is a sub-concept of `Umwelt`. The respective WSML code is written in Listing 2.

Listing 2 The concepts `Activity` and `ActivityUmwelt` with attributes connecting each other and linking the `Activity` to an `Agent`. The cardinality constraints make sure that one `Activity` is linked to exactly one `Agent` and that one `ActivityUmwelt` is linked to exactly one `Activity`.

```
concept Activity subConceptOf Entity
    hasUmwelt ofType (1 1) ActivityUmwelt
    hasAgent ofType (1 1) Agent

concept ActivityUmwelt subConceptOf Umwelt
    hasActivity ofType (1 1) Activity
```

To *glue* our concepts together and to ensure that the models of our ontology comply with the informal theory we introduce axioms. We have for example an axiom that states that if an `Agent` has an `Activity` with an `ActivityUmwelt` this `ActivityUmwelt` is an `Umwelt` of the `Agent`. The WSML code for this axiom is given in Listing 3. Analogously, we defined an axiom to enforce the link between `Agent` and `Activity` in case the `Agent` has the `ActivityUmwelt` for that `Activity`.

In a similar fashion, we implemented the concepts `Object`, `Event`, `State`, two additional sub-concepts of `Umwelt`, namely `UniversalUmwelt` and `CompoundActivityUmwelt`, as well as `CompoundActivity`.

Listing 3 Axiom A4 of our ontology stating that if `Agent ?x` has an `Activity a` and the `Activity ?a` has an `ActivityUmwelt ?u`, then the `Agent ?a` also has that `ActivityUmwelt ?u`.

```
axiom A4
  definedBy
    ?x memberOf Agent and
    ?u memberOf ActivityUmwelt and
    ?a memberOf Activity and
    ?x[hasActivity hasValue ?a] and
    ?a[hasUmwelt hasValue ?u] implies
    ?x[hasUmwelt hasValue ?u].
```

The mereological structures that are one pillar of our theory are implemented as WSML relations `partOf` and its inverse `hasPart`. The transitivity, reflexivity, and antisymmetry of the parthood relations are formalized in axioms. In particular, our implementation represents objects as parts of Umwelten, Activity Umwelten as potential parts of Activity Umwelten or Compound Activity Umwelten, and Activity Umwelten as part of a Universal Umwelt. Furthermore, the compound activity has activities as parts. An illustration of all the concepts and their attributes is shown in Figure 5. The dotted links in Figure 5 are not implemented in the ontology but should be implemented as axioms in applications of this ontology.

5 Practical Evaluation of the Umwelt Theory

In this section, we give a quick overview of the architecture we used before showing in more detail how our Umwelt ontology can be employed to implement our scenario. The implementation includes an extension of our Umwelt ontology with concepts and instances from our domain of application. After that, we present the integration of the extended Umwelt ontology into an Agent Based Simulation. Finally, we visualize a simulation of the notification scenario from Section 1. The Agent Based Simulation allows us to test our ontology without setting up a whole infrastructure of Web services.

5.1 Architecture of our Prototype

For the scenario we have imported the necessary components of our architecture into the Repast tool-kit[8]. Repast is open-source software providing a GIS interface and support for different programming languages, e.g., Java. Therefore, it is possible to integrate the WSMO4J[9] and the wsml2reasoner[10] Java libraries. These libraries enable the creation and handling of WMSL ontologies in Java as well as the execution of reasoning processes. Figure 6 shows a diagram of the components of our prototype.

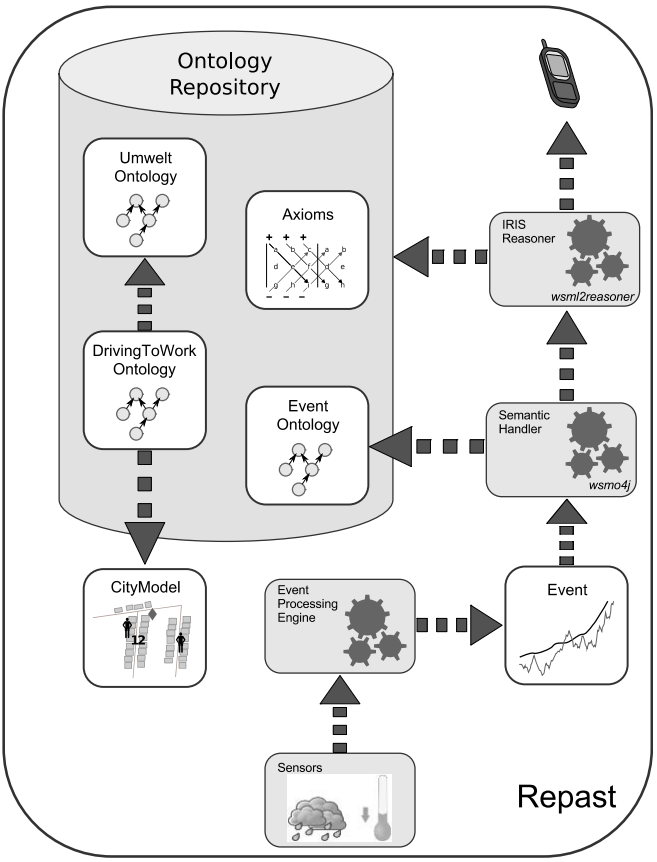


Fig. 6. The architecture of our prototype.

[8] Repast can be downloaded from http://repast.sourceforge.net/ (last accessed 15.12.2010), more information is published in [20].

[9] Available at http://wsmo4j.sourceforge.net/ (last accessed 15.12.2010)

[10] Available at http://tools.sti-innsbruck.at/wsml2reasoner/ (last accessed 15.12.2010)

At the core of our architecture is the Ontology Repository. The Ontology Repository is an ontology management system (comparable to Database Management Systems) that stores our ontologies and axioms. We store three ontologies. We base our prototype on the Umwelt Ontology (cf. Section 4), which is extended by a DrivingToWork Ontology. The DrivingToWork Ontology reflects a city model of a city in which our agents live. The axioms constraining the content of all ontologies, i.e. rules that specify for example which agent is affected by which event through the activity, are stored within the ontologies to which they belong. The Event Ontology specifies the events that occur in the city, for example a rainfall event or a freezing event. These events are detected by the Event Processing Engine, which parses sensor observations. At the moment, we use a simple Event Processing Engine that detects events according to predefined thresholds in the streamed observation values. If an event is detected, it is sent to the Semantic Handler. The Semantic Handler generates instances of events in the Event Ontology using the WSMO4J library during runtime of the Simulation. Finally, the Semantic Handler triggers an IRIS Reasoner[11] (via the wsml2reasoner library) every-time an event is generated. The reasoner checks the axioms and sends a notification to the agents in case they subscribed to the event that triggered the reasoning. Our system constitutes a very simple context-sensitive system as in [12], mapping from quantitative values into qualitative categories, and alerting agents according to their individual needs (expressed through subscription to predefined events). We have written simple implementations of the Java Components ourselves, but will not discuss them here as they are trivial and only developed as far as to realize our scenario. However, our work, i.e. sourcecodes, axioms, and ontologies, is available on the web[12].

5.2 Domain-Specific Extension of the Umwelt Ontology

The Umwelt ontology provides the necessary backbone to model hierarchies of environments of agents and the impacts on them. A designer creates domain concepts and links them to the provided upper-level structure. When all domain concepts are created, it is possible to generate instances of them. Instances are members of concepts. They represent unique objects of the real world using the identification criteria defined in the concepts. For example, a concept `Building` (domain concept) is a sub-concept of

[11] More information is published on http://www.iris-reasoner.org/ (last accessed 15.12.2010)

[12] Visit http://trac6.assembla.com/soray/wiki/SwoyAgile to find links to the ontologies used in the work presented here and http://trac6.assembla.com/soray/wiki/SwoyABMDownloads to download Repast and a project with our simulation (both accessed 22.12.2010)

PhysicalObject (upper-level concept). The representation of an individual entity and not of a category is an instance in our ontology. For example, the representation of a building having a specific address value is a concrete instance of the concept Building. This instance represents a single real world building.

Following these suggestions we created the domain concepts required for the scenario. First, the activity concept called DrivingToWorkByBike was generated. This is an activity, which is performed by many people. To describe one activity of one person, it needs properties to make it unique. We identified two criteria of uniqueness, the time span (given by start point and end point) and the agent executing it. The scaling of the activity is up to the designer, so we decided to model DrivingToWorkByBike as an activity. The activity has an Umwelt represented as DrivingToWorkByBikeUmwelt. In our case, we model this Umwelt containing only roads as an object. If the RepastCityRoad changes its State to Icy or Flooded, the Activity is affected. A road can be identified through two coordinates representing start- and endpoint, and its name. Our RepastCityRoad can have the States Passable, Icy, Wet, Flooded, and Closed. When the roads used by John or other citizens participating in the mentioned activity will become icy or flooded they will get a notification including the relevant roads and their states. In Listing 4, our domain concepts are listed.

Listing 4 The concepts represent the domain of our scenario. They are linked to the concepts of our Umwelt ontology.

```
concept DrivingToWorkByBike subConceptOf Activity
    hasUmwelt ofType DrivingToWorkByBikeUmwelt
    hasAgent ofType RepastCityAgent

concept DrivingToWorkByBikeUmwelt subConceptOf ActivityUmwelt
    hasPart ofType RepastCityRoad
    hasActivity ofType DrivingToWorkByBike

concept RepastCityRoad subConceptOf Object
    partOf ofType DrivingToWorkByBikeEnvironment
    hasState ofType State

concept RepastCityAgent subConceptOf Agent
    hasActivity ofType DrivingToWorkByBike

concept RepastCityObserver subConceptOf Observer
    hasObserverUmwelt ofType ObserverUmwelt

concept ObserverUmwelt subConceptOf UniversalUmwelt
    contains ofType (1 *) RepastCityRoad
concept Icy subConceptOf State
```

```
concept Flooded subConceptOf State
```

Listing 5 The event `IcyRoads` is described in WSML. The `IcyRoads` event is triggered by an amount of precipitation greater than 0mm/h and a ground temperature lower than 0°C.

```
axiom IcyRoads
  definedBy
    ?notifiedEvent memberOf dolce#Event and
    ?precipitationElement[
        dolce#participantIn hasValue ?precipitation,
        dolce#hasQuality hasValue
    ?ground[
        dolce#participantIn hasValue ?heatFlow,
        dolce#hasQuality hasValue ?temp] memberOf G
        ?precipitation memberOf Precipitation and
        ?heatFlow memberOf HeatFlow and
        ?precipitationAmount [value hasValue ?value1] mem-
berOf PrecipitationAmount and
        ?temp [value hasValue ?value2] memberOf Temperature
and
        ?value1 > 0 and
        ?value2 < 0 implies ?notifiedEvent memberOf IcyRoads.
```

In the assumed case, there is a knowledge engineer responsible for creating concepts and models of natural events like icy roads. As proof of concept, we created two basic events for icy roads and heavy rainfall. The sourcecode for the icy roads event is given in Listing 5.

5.3 Instantiation and Simulation

To create instances, simulate events, and perform reasoning, we use Agent Based Simulation (ABS) [24] with the Repast tool-kit. ABS enables the simulation of actions of agents with behaviour in an environment. For our scenario, we used the RepastCity model [15]. RepastCity provides shapefiles of a city model including roads, buildings, and moving agents that are instantiated as WSML instances and stored in an ontology repository.

For every `Agent` instance, an `Activity` instance is created being a member of the `DrivingToWorkByBike`concept. All `Activity` instances are linked to `ActivityUmwelt` instances, which again are linked to `Road` instances using the `partOf` relation. Therefore, every `Agent` is connected to all `Road` instances of his route to work via its `Activity` and its `Umwelt`. Furthermore, we extended the RepastCity model with some sensors modelled as non-moving agents to simulate the observations. The observation values (temperature and precipitation) are

used to instantiate the introduced events. Figure 7 shows the initiated city model.

Fig. 7. The city model after the initialization. The sensors are displayed as grey rhombi. Our 14 agents are depicted as numbered icons. We have adjusted the illustration of colours and icons from repast to increase clarity.

We can integrate the Umwelt ontology with its domain-specific extension into the Agent Based Simulation. This allows us to simulate the behaviour of our ontology in a dynamic real world scenario with live sensor observations and notifications.

We run a simulation of our scenario with a total number of 14 agents. These agents have predefined home places. The working places are randomly determined during the initialization phase. Furthermore, there are 24 sensors producing temperature and precipitation values. For demonstration purposes we randomly increase or decrease the values within a predefined interval in every step of the simulation. The Event Processing Engine takes the sensor observation values as input and compares them to predefined threshold values modelling the occurrence of an event. In a second step, the Semantic Handler receives a notification about the occurrence of

an event detected for a certain sensor. The Semantic Handler, generates an instance for this event in our event ontology. The introduction of a new event triggers the execution of axioms that change the state of affected roads in the DrivingToWork ontology according to the new condition. Subsequently, the Reasoning Engine queries all agents that are currently using roads whose state is `icy` or `flooded`. The query engine exploits the links between agents, their activities and their Umwelt that is modelled in our Umwelt ontology. The queried agents are notified and change their behaviour. In our case, the simulated agents move slower if an event occurs and affects their activity. This behaviour is implemented in this way to show that the cycling agent is now walking and also to indicate that he received a notification.

6 Future Work

The presented work leaves opportunities for future work. On the one hand, by extending the theory itself, on the other hand, by developing further applications and integrating them into larger knowledge infrastructures.

Possible extensions of the theory are at first an extension of the Umwelt theory towards topology and location as done for the Niche [28, 29]. A distinction of environments based on proximity to the agent was suggested for example in [33], [11], and more recently in [18]. Furthermore, we have not yet made temporal constraints explicit. It is highly desirable to anchor Umwelten in a full space-time model, for example as suggested in [22] for location-based services. The exact structure of Umwelten is not yet fully explored. We are also going to connect the Umwelt theory to our previous work on affordance in ontology [21].

Even though not necessary in our application, the formal theory would also benefit from an alignment to a foundational ontology, as for example the Descriptive Ontology of Linguistic and Cognitive Engineering [16]. Especially concepts like `Event` and `Object`, which are not analysed in this paper, can be set on a solid basis. Moreover, the alignment to a foundational ontology facilitates the integration with other ontologies. For example, our approach presumes that a provider takes care of modelling activities and events. An approach how ontologies of activities can be automatically generated is given in [13].

The implementation is demonstrated as agent based simulation, for future work we would like to use standardized Web services instead of simulated input and output channels. We plan to get real world data provided by Sensor Observation Services (SOS) [19] and send notifications via Sen-

sor Alert Services (SAS) [27]. With this, we would make a practical contribution to the Semantic Sensor Web (SSW) [26]. An approach for a Web-based information infrastructure using the Umwelt theory is presented in [17]. Another interesting emerging topic is the exploitation of human observations [10] for applications like notification services. Additionally, we modelled the events based on naïve assumptions. For real world applications, we would also like to consult domain experts and create better models of events.

7 Conclusion

We have presented a theory of Umwelten of agents. The Umwelt consists of objects, which are changed by events in the Umwelt. The theory of Umwelten provides a functional and dynamic account of activities in geographic spaces. Formally, the theory is based on mereology. The implementation is done in WSML, which facilitates the integration of Web services, like Sensor Observation Services. Therefore, we can easily use the theory as ontology in information infrastructures. To show that our theory can be applied in solving our scenario of driving to work, we have implemented a small notification infrastructure and executed it as Agent Based Simulation. The Repast toolkit served our needs well and it was possible to model how our ontology, different components of the infrastructure, and sensors interact. We found that our approach was able to solve our scenario, that is we notified agents in case of freezing or flooding events, and the agents in our simulations reacted according to this notification. The scenario and the set up of our simulation are rather simple, yet we do not see big problems in extending the simulation to be more realistic in terms of more sensors, observation of further properties, modelling of events in more detail, and a larger variety of events, as well as taking into account the requirements of different agents. We see a great potential of our ontology to provide more precise notifications of environmental phenomena, in general. Providing notification services via Web portals can allow non-expert users to easily retrieve live information according to their specific temporal, spatial, and thematic constraints.

Acknowledgement

This work has been partly funded through the International Research Training Group on Semantic Integration of Geospatial Information by the

DFG (German Research Foundation, GRK 1498) and partly by the European research project ENVISION (FP7 249170).

References

1. R.G. Barker. Ecological psychology: Concepts and methods for studying the environment of human behavior. Stanford University Press, Stanford, CA, 1968.
2. B. Bennett. Foundations for an Ontology of Environment and Habitat. In Galton, A. and Mizoguchi, R., editor, Formal Ontology in Information Systems Proceedings of the Sixth International Conference (FOIS 2010), volume 209 of Frontiers in Artificial Intelligence and Applications, pages 31–44, Amsterdam Berlin Tokyo Washington, DC, 2010. IOS Press.
3. M. Botts, G. Percivall, C. Reed, and J. Davidson. OGC® Sensor Web Enablement: Overview and High Level Architecture. OpenGIS® White Paper OGC 07-165, Open Geospatial Consortium Inc., December 2007.
4. A. Chemero. An outline of a theory of affordances. Ecological Psychology, 15(2):181–195, 2003.
5. J. de Bruin, D. Fensel, U. Keller, M. Kifer, H. Lausen, R. Krummenacher, A. Polleres, and L. Predoiu. Web Service Modeling Language (WSML). W3c member submission, World Wide Web Consortium (W3C), June 2005.
6. J. Deely. Umwelt. Semiotica, 2001(134):125–135, 2001.
7. A. Galton and M. Worboys. Processes and events in dynamic geo-networks. In M. Rodríguez, Isabel Cruz, Sergei Levashkin, and Max Egenhofer, editors, GeoSpatial Semantics, volume 3799 of Lecture Notes in Computer Science, pages 45–59. Springer Berlin / Heidelberg, 2005.
8. J.J. Gibson. Perceiving, acting, and knowing: Toward an ecological psychology, chapter The Theory of Affordances, pages 67–82. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, 1977.
9. J.J. Gibson. The ecological approach to visual perception. Lawrence Erlbaum, Hillsdale, NJ, 1979.
10. M.F. Goodchild. Citizens as sensors: the world of volunteered geography. GeoJournal, 69(4):211–221, 2007.
11. J.G. Granö. Pure geography, 1997. Originally published as Reine Geographie by the Geographical Society of Finland in Acta Geographica, vol 2, 1929, and as Puhdas maantiede by Werner Sönderström, Porvoo, 1930.
12. C. Keßler, M. Raubal, and C. Wosniok. Semantic rules for context-aware geographical information retrieval. In Proceedings of the 4th European conference on Smart sensing and context, pages 77–92. Springer-Verlag, 2009.
13. W. Kuhn. Ontologies in support of activities in geographical space. International Journal of Geographical Information Science, 15(7):613–631, 2001.
14. W.Kuhn. Research Trends in Geographic Information Science, chapter Semantic Engineering, pages 63–76. Lecture Notes in Geoinformation and Cartography. Springer-Verlag, Berlin Heidelberg, 2009.

15. N. Malleson. RepastCity – A Demo Virtual City, 2008. Available online at http://portal.ncess.ac.uk/access/wiki/site/mass/repastcity.html.
16. C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Wonderweb deliverable D18 ontology library (final). ICT Project, 33052, 2003.
17. H. Michels and P. Maué. Semantics for notifying events in the affecting environment. In K. Greve and A.B. Cremers, editors, EnviroInfo 2010, Proceedings of the 24th International Conference on Informatics for Environmental Protection, Cologne/Bonn, Germany, pages 501–507, Aachen, Germany, 2010. Shaker Verlag.
18. D.Montello.Scale and Multiple Psychologies of Space. In I. Campari and A.U. Frank, editors, Spatial Information Theory: A Theoretical Basis for GIS, International Conference COSIT '93, Marciana Marina, Elba Island, Italy, September 19-22, 1993, Proceedings, volume 716 of Lecture Notes in Computer Science, pages 312–321, Heidelberg, 1993. Springer.
19. A. Na and M. Priest. Sensor Observation Service. OpenGIS® Implementation Standard OGC 06-009r6, Open Geospatial Consortium Inc., October 2007.
20. M.J. North, N.T. Collier, and J.R. Vos. Experiences creating three implementations of the repast agent modeling toolkit. ACM Transactions on Modeling and Computer Simulation (TOMACS), 16(1):1–25, January 2006.
21. J. Ortmann and W. Kuhn. Affordances as Qualities. In Galton, A. and Mizoguchi, R., editor, Formal Ontology in Information Systems Proceedings of the Sixth International Conference (FOIS 2010), volume 209 of Frontiers in Artificial Intelligence and Applications, pages 117–130, Amsterdam Berlin Tokyo Washington, DC, May 2010. IOS Press.
22. M. Raubal, H.J. Miller, and S. Bridwell. User-Centred Time Geography for Location-Based Services. Geografiska Annaler: Series B, Human Geography, 86(4):245–265, 2004.
23. S.J. Russell, P. Norvig, J.F. Canny, J. Malik, and D.D. Edwards. Artificial intelligence: a modern approach. Prentice Hall, Englewood Cliffs, NJ, 1995.
24. D.A. Samuelson and C.M. Macal. Agent-based simulation comes of age. OR/MS TODAY, 33(4):34–38, August 2006.
25. R.E. Shaw, M.T. Turvey, and W. Mace. Cognition and the Symbolic Processes, volume 2, chapter Ecological Psychology: The Consequence of a Commitment to Realism, pages 159–226. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1982.
26. A Sheth, C Henson, and SS Sahoo. Semantic Sensor Web. IEEE Internet Computing, 12(4):78–83, 2008.
27. I. Simonis and J. Echterhoff. OGC® Sensor Alert Service Implementation Specification. Candidate OpenGIS® Interface Standard OGC 06-028r5, Open Geospatial Consortium Inc., May 2007.
28. B. Smith and A.C. Varzi. The Niche. Nous, pages 214–238, 1999.
29. B. Smith and A.C. Varzi. Surrounding Space: An Ontology of Organism-Environment Relations. Theory in Biosciences, 121(2):139–162, 2002.
30. N. Steinmetz and I. Toma. D16.1 v1.0 WSML Language Reference. WSML Final Draft 2008-08-08, 2008. http://www.wsmo.org/TR/d16/d16.1/v1.0/20080808/.

31. T.A. Stoffregen. Affordances as Properties of the Animal-Environment System. Ecological Psychology, 15(2):115–134, 2003.
32. M.T. Turvey. Affordances and prospective control: An outline of the ontology. Ecological Psychology, 4(3):173–187, 1992.
33. J. von Uexküll and G. Kriszat. Streifzüge durch die Umwelten von Tieren und Menschen: ein Bilderbuch unsichtbarer Welten – Bedeutungslehre. Rowohlt, Hamburg, 1956. Originally published in 1934 in Sammlung Verständliche Wissenschaft, Berlin.

Spatial Data Processing and Structuring

Detecting Symmetries in Building Footprints by String Matching

Jan-Henrik Haunert

Chair of Computer Science I, University of Würzburg, Germany
jan.haunert@uni-wuerzburg.de

Abstract. This paper presents an algorithmic approach to the problem of finding symmetries in building footprints. The problem is motivated by map generalization tasks, for example, symmetry-preserving building simplification and symmetry-aware grouping and aggregation. Moreover, symmetries in building footprints may be used for landmark selection and building classification.

The presented method builds up on existing methods for symmetry detection in polygons that use algorithms for string matching. It detects both axial symmetries and repetitions of geometric structures. In addition to the existing string-matching approaches to symmetry detection, we consider the problem of finding partial symmetries in polygons while allowing for small geometric errors. Moreover, we discuss how to find optimally adjusted mirror axes and to assess the quality of a detected mirror axis using a least-squares approach.

The presented approach was tested on a large building data set of the metropolitan Boston area. The dominant symmetry relations were found. Future work is needed to aggregate the obtained symmetry relations, for example, by finding sets of mirror axes that are almost collinear. Another open problem is the integration of information on symmetry relations into algorithms for map generalization.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_16,

1 Introduction

Many buildings contain symmetric structures. No matter whether a symmetric building design was chosen for aesthetics, functionality, or simply for minimizing construction costs, humans perceive symmetry as an important building characteristic. Since many geographic analysis tasks require methods for shape characterization, an automatic symmetry detector is needed. This paper presents a new algorithm for the detection of symmetries in polygons. This algorithm is tailored to deal with building footprints that we typically find in cadastral or topographic databases. It was tested for a building data set of the metropolitan Boston area.

A building footprint consists of multiple polygonal rings (that is, one exterior ring and multiple interior rings). The presented method finds (partial) symmetries in one ring or between two rings, regardless of whether the rings are interior or exterior. In the following, we refer to each ring as a polygon.

The work presented in this paper contributes to the general aim of enriching spatial data with information on geometric structures and patterns. Such information is valuable for multiple applications. Our main motivation is *map generalization*, which aims to decrease a map's level of detail while preserving its characteristic structures. With respect to buildings, we particularly aim at symmetry-preserving simplification and symmetry-aware aggregation. Both problems have not been approached yet.

For *building simplification* we recently presented an algorithm based on discrete optimization (Haunert and Wolff 2010). This algorithm allows us to integrate multiple quality criteria such as the preservation of a building's area and its dominating edge directions. Symmetry preservation, however, is currently not considered as a quality criterion in our method (and not in other methods), thus we may lose symmetric structures by simplification (see Figure 1). In order to overcome this drawback, we need to detect symmetries in the input building. Then, we can define a cost function that penalizes those simplifications that destroy symmetries.

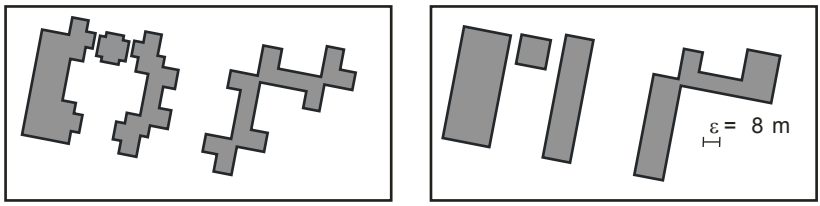


Fig. 1. Two buildings (left) and their simplifications (right) obtained with the building simplification method by Haunert and Wolff (2010) and the error tolerance $\varepsilon = 8$ m. The symmetry relations are lost.

Building aggregation means to find groups of buildings. Each group may be replaced by a single map object, for example, a building block. In map generalization, the grouping of objects is usually done according to Gestalt criteria, for example, alignment, similarity, and proximity, which model criteria of human perceptional grouping (Wertheimer 1938). Obviously, symmetry is an important criterion for grouping. In Figure 2, we clearly perceive that five buildings form an ensemble – this is because of their symmetric arrangement. Therefore, replacing the ensemble by a single shape can be a favorable generalization action.

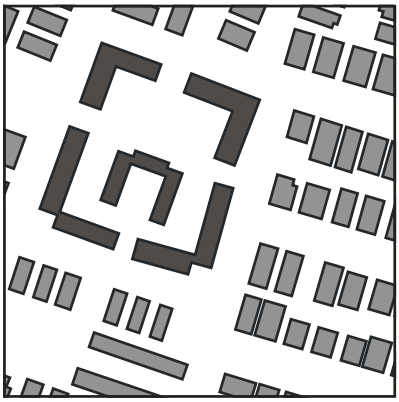


Fig. 2. An ensemble of five buildings (dark grey) that a human can identify based on symmetry.

Fig. 3. Because of symmetry relations, the dark grey building can be used as a landmark.

Map generalization is not the only application of symmetry detection. For example, buildings whose major symmetry axes are collinear with important lines of sights can serve as landmarks for navigation (see Figure 3). Moreover, such buildings often have representative functions like town halls or castles. The dark grey building in Figure 3, for example, is the main building of Harvard Medical School. Therefore, symmetry can be used as a cue for both *automatic landmark selection* (that is, deciding which building serves best as a landmark in a routing instruction) and *building classification*, which are topical problems in geographic information science. For a recent approach to compare different landmark selection methods we refer to Peters et al. (2010). Steiniger et al. (2008) and Werder et al. (2010) have proposed shape measures to classify building footprints and, more generally, polygons according to their functionality.

The paper is structured as follows. We first discuss related work on data enrichment in map generalization and on algorithms for symmetry detection (Section 2). Section 3 introduces a new algorithm for symmetry detec-

tion. In Section 4, we discuss experimental results with this algorithm. Section 5 concludes the paper.

2 Related Work

The gathering of knowledge on patterns and structures in geographic data, *data enrichment*, is often considered a prerequisite for automatic map generalization (Mackaness and Edwards 2002; Neun et al. 2008; Steiniger 2007). Thomson and Brooks (2002) show how to find long sequences of (almost) collinear road segments in road datasets. Such sequences, so-called *strokes*, correspond to major road axes that need to be preserved during generalization. Heinzle and Anders (2007) present algorithms to find star-like structures, rings, and regular grids in road networks in order to improve the generalization of networks. Christophe and Ruas (2002), as well as Ruas and Holzapfel (2003), present methods to find alignments of buildings. Gaffuri and Trévisan (2004) show how to deal with such patterns in a multi-agent system for map generalization. Methods for the grouping of buildings are proposed by Regnauld (2003) and Yan et al. (2008). These methods, however, do not consider symmetry as a criterion for grouping.

In contrast, symmetry detection has found much attention in the literature on image analysis and pattern recognition. Symmetry detection in images is often done based on local image features that are highly distinctive and invariant against certain transformations, for example, rotation and scale. Loy and Eklundh (2006) as well as Cho and Lee (2009), for example, use so-called SIFT (scale-invariant feature transform) descriptors. A comparative study on symmetry detection in images is given by Park et al. (2008). Mitra et al. (2006) present a method for finding symmetries in three-dimensional models. Similar to the symmetry detectors for images, their method relies on characteristic points. In this case, however, these points are defined based on the curvature of the model's surface. Point pairs that correspond by shape symmetry are found using RANdom SAmple Consensus (RANSAC).

In contrast to symmetry detection in images, symmetry detection in two-dimensional polygons is often done by string matching. The basic string matching approach of Wolter et al. (1985) is to encode the polygon P as a string X, for example, as a sequence of angles and edge lengths, see Figure 4. In order to find an axial symmetry relation, we need to test whether the string X^{-1} (meaning the reversal of X) is a substring of the string XX (meaning the concatenation of X with itself). This test can be done in $\Theta(n)$

time where n is the number of elements in XX by using the algorithm of Knuth et al. (1977). In the example in Figure 4, the string X^{-1} is indeed a substring of XX. Its location within XX yields the axial symmetry relation. Similarly, we can find a rotational symmetry relation by finding X itself within XX. We need to avoid trivial solutions, however, that match X to the first or second half of XX. This can be done by removing the first and the last element from XX before matching. Based on a similar approach by string matching, the algorithm of Atallah (1985) finds *all* axes of symmetry of a polygon with n vertices in $\Theta(n \log n)$ time. In order to cope with geometric distortions, Lladós et al. (1997) use an approach based on a string edit distance.

Yang et al. (2008) present an approach to symmetry detection based on critical contour points. The critical points are the vertices of a simplified version of the original contour. However, since symmetry-preserving algorithms for line and building simplification do not exist, we need to be careful with this approach. In the preprocessing, we use a building simplification algorithm with a conservative setting in order to remove marginal but potentially disturbing details.

The general string-matching approach seems to be applicable for symmetry detection in building footprints. Not considered in the string-matching approaches discussed, however, is the problem of finding *partial* symmetry relations (only parts of the shape are symmetric). The next section presents a solution to this problem. Furthermore, we address the problem of generating optimal mirror axes by least-squares adjustment.

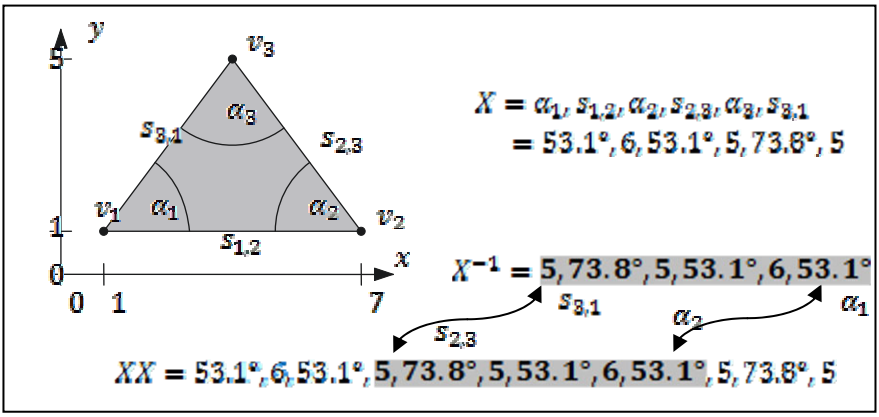


Fig. 4. Principle of the algorithm for symmetry detection by Wolter et al. (1985). By finding string X^{-1} in string XX, it becomes clear that the polygon has an axial symmetry relation. According to that relation, for example, edge $s_{3,1}$ is a mirror image of edge $s_{2,3}$.

3 Methodology for Symmetry Detection

Generally, the symmetry relations we aim to detect are geometric transformations of which each map is a continuous part p_1 of a building outline onto (or sufficiently close to) a continuous part p_2 of a second building outline. Both parts may either belong to different polygons or to the same polygon. Moreover, both p_1 and p_2 may be the same. For instance, let p_1 and p_2 be equal to the entire polygon in Figure 4. Indeed, there is a non-trivial transformation that maps p_1 onto itself: the reflection at the vertical line through v_3. Reflections, however, are but one type of transformation we can detect with the presented method. More generally, we allow the following two types of transformations:

[1] p_2 is obtained by (successively) translating and rotating p_1

[2] p_2 is obtained by (successively) reflecting, translating, and rotating p_1.

Accordingly, we term the pair (p_1, p_2) a *type-1 match* or a *type-2 match*. In particular, we are interested in axial symmetries, that is, type-2 matches that correspond by a reflection on a straight line.

We first formalize the problems of finding type-1 and type-2 matches as a string matching problem (Section 3.1) and then discuss a solution by dynamic programming (Section 3.2). Finally, we discuss an approach based on least-squares adjustment that allows us to find axial symmetries in the detected set of type-2 matches (Section 3.3).

3.1 Symmetry Relations in the String Representation

By encoding a polygon P as a string $X(P)$ of edge lengths and angles, we obtain a shape representation that is invariant against rotations and translations. This allows us to define each type-1 match based on a pair of *similar* strings, one of them being a substring of $X(P_1)X(P_1)$ and the other one a substring of $X(P_2)X(P_2)$, where P_1 and P_2 are two potentially distinct polygons. Similarly, we define each type-2 match based on a pair of similar strings, one of them being a substring of $X(P_1)X(P_1)$ and the other one a substring of $X^{-1}(P_2)X^{-1}(P_2)$. Two strings x_1 and x_2 are called similar if the following four criteria hold:

[1] The number k of symbols is the same in both strings.

[2] Both strings start with the same type of symbol, that is, either with a symbol representing an edge length or an angle.

[3] For $i = 1,2, \ldots, k$, if the i-th symbol in x_1 and the i-th symbol in x_2 represent angles, both angles differ at most by $\Delta\alpha_{max}$.

[4] For $i = 1,2, \ldots, k$, if the i-th symbol $x_1(i)$ in x_1 and the i-th symbol $x_2(i)$ in x_2 represent edge lengths, the ratio $\max\{x_1(i), x_2(i)\}/\min\{x_1(i), x_2(i)\}$ does not exceed $1 + \Delta l_{max}$.

The parameters $\Delta\alpha_{max} \in \mathbb{R}_0^+$ and $\Delta l_{max} \in \mathbb{R}_0^+$ allow users to specify the geometric error tolerance. Furthermore, we define the number k of symbols as the *cardinality* of a match. We are not interested in matches of single line segments, which have cardinality one. In order to exclude such insignificant matches, a user needs to define a third parameter $k_{min} \in \mathbb{N}$. The cardinality of a match must not be smaller than k_{min}.

Next, we exclude matches that are *dominated* by other matches: A match of two strings x_1 and x_2 is dominated by a match of two strings y_1 and y_2 if

- x_1 is a substring of y_1 and x_2 is a substring of y_2 and
- x_1 has the same position in y_1 as x_2 in y_2, that is, the number of symbols in y_1 preceding x_1 equals the number symbols in y_2 preceding x_2.

Additionally, we need to take care that we do not select a substring of a string XX that is longer than the original string X representing the polygon and we should avoid reporting a match of two polygon parts twice.

Finally, if we have found a match of two strings that satisfies the above-mentioned criteria, we need to decode the two strings into two shapes, for example, to visualize the matching result. The shapes p_1 and p_2 for the two strings x_1 and x_2 of a match are computed as follows.

For each edge symbol in a string, we add the corresponding polygon edge to the shape for the string. If the string begins (or ends) with a symbol for an angle, we add both polygon edges that form this angle. With this approach, however, the first (or last) edge of p_1 and the first (or last) edge of p_2 get very different lengths. Therefore, we shorten the longer edge of both unmatched edges such that they get the same lengths.

3.2 String Matching by Dynamic Programming

In this section we discuss a solution to the problem of finding all type-2 matches satisfying the criteria from Sect. 3.1. The type-1 matches can be found in a straightforward way. We first discuss the special case that $\Delta\alpha_{max} = \Delta l_{max} = 0$. In this case, a type-2 match of maximum cardinality

can be found by solving the *longest (or maximum) common substring problem* for the strings $X(P_1)X(P_1)$ and $X^{-1}(P_2)X^{-1}(P_2)$.

The longest common substring problem can be solved in linear time using a generalized suffix tree (Gusfield 1997). We are interested, however, in finding multiple symmetry relations. Therefore, we search for *all maximal common substrings* of $X(P_1)X(P_1)$ and $X^{-1}(P_2)X^{-1}(P_2)$. Note that there is a difference between a maximum and a maximal common substring of two strings x_1 and x_2: a common substring x of x_1 and x_2 is *maximum* if no other common substring of x_1 and x_2 is longer than x; for x being a *maximal* common substring, however, it suffices that there is no other common substring of x_1 and x_2 that contains x, that is, a match defined by a maximal common substring is not dominated by any other match.

The problem of finding all maximal common substrings of two strings x_1 with m symbols and x_2 with n symbols can be solved in $\Theta(mn)$ time by dynamic programming. To specify this approach, we define the $m \times n$ matrix D of integers. We denote the number in row i and column j of D by $d_{i,j}$. Additionally, we define $d_{0,j} = d_{m+1,j} = 0$ for $j = 0,1,\ldots,n+1$ and $d_{i,0} = d_{i,n+1} = 0$ for $i = 0,1,\ldots,m+1$. For $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$ we define

$$d_{i,j} = \begin{cases} 1 + d_{i-1,j-1} & \text{if } x_1(i) = x_2(j) \\ 0 & \text{else} \end{cases}, \qquad (1)$$

where $x_1(i)$ denotes the i-th symbol in x_1 and $x_2(j)$ the j-th symbol in x_2. The values of D can be computed in increasing order of the indices for rows and columns.

Once we have computed the matrix D, we can easily find the maximal common substrings. For each pair $i \in \{1,2,\ldots,m\}$, $j \in \{1,2,\ldots,n\}$ with $d_{i,j} > 0$ and $d_{i+1,j+1} = 0$, the substring of x_1 starting at index position $(i - d_{i,j} + 1)$ and ending at index position i corresponds to one maximal common substring of x_1 and x_2. In x_2, this substring starts at index position $(j - d_{i,j} + 1)$ and ends at index position j.

In order to deal with geometric differences between the two building parts of a match and to avoid the selection of substrings that are longer than the original encoding of the building polygon, we define the values $d_{i,j}$ in a slightly different way:

$$d_{i,j} = \begin{cases} 1 + d_{i-1,j-1} & \text{if } x_1(i) \approx x_2(j) \text{ and } d_{i-1,j-1} < \min\{m/2, n/2\}, \\ 1 & \text{if } x_1(i) \approx x_2(j) \text{ and } d_{i-1,j-1} = \min\{m/2, n/2\}, \\ 0 & \text{else} \end{cases} \quad (2)$$

We define the relation $\approx$ according to the similarity criteria 3 and 4 that we introduced in Sect. 3.1. The additional condition $d_{i-1,j-1} < \min\{m/2, n/2\}$ in the first line of equation (2) avoids that we generate strings that are too long, that is, if $d_{i-1,j-1}$ is equal to the length of the string for one of the involved polygons, we do not further extend the corresponding match but start with the construction of a new match. This is done in the second line of equation (2) by setting $d_{i,j}$ to one.

In order to avoid reporting the same match twice, we need to introduce a small modification to the procedure for finding the maximal common substrings in D: instead of considering each pair of indices $i \in \{1,2,\ldots,m\}$, $j \in \{1,2,\ldots,n\}$ for defining the two ends of the corresponding substrings, we only consider each pair of indices $i \in \{m/2 - 1, m/2, \ldots, m-1\}$, $j \in \{n/2 - 1, n/2, \ldots, n-1\}$.

Note that, when implementing the presented method, we should avoid comparing edges with angles. Therefore, we can use two matrices D_α and D_e, each of dimension $m/2 \times n/2$, instead of one matrix D of dimension $m \times n$. We use D_α for the comparisons of angles and D_e for the comparisons of edge lengths.

3.3 Least-Squares Adjustment

As a result of the algorithm in Section 3.2 we obtain a set of matches, each represented as a pair of strings. We can use the decoding presented in Section 3.1 to find the corresponding pair of shapes. The two shapes p_1 and p_2 of a match are polylines, both having the same number κ of vertices. For $i = 1,2,\ldots,\kappa$, the i-th vertex of p_1 corresponds with the i-th vertex of p_2.

If p_1 and p_2 correspond by axial symmetry, we can compute the mirror axis by choosing any pair of corresponding vertices v_1 and v_2 and computing a straight line that is perpendicular to the vector $\overrightarrow{v_1 v_2}$ and passes through the midpoint between v_1 and v_2. If we do this for each type-2 match, we obtain candidates for mirror axes. These axes, however, are not very accurate, because we used a single pair of vertices for their construction. In order to obtain more accurate mirror axes, we apply a least-squares adjustment that uses the information given with *all* pairs of corresponding

vertices. The main benefit of this approach is that, in addition to the adjusted mirror axis, it offers a standard deviation that allows us to conclude whether the match indeed corresponds to an axial symmetry, or whether another type of transformation is involved, for example, a transformation or rotation.

For the adjustment we use a Gauss-Helmert model, which has the general form

$$\Psi(\underline{\tilde{X}}, \underline{\tilde{L}}) = \underline{0}, \tag{3}$$

where $\underline{\tilde{X}}$ is the vector of unknowns without errors and $\underline{\tilde{L}}$ the vector of observations without errors (Niemeier 2002). The aim of the adjustment process is to add a vector $\underline{v}$ of corrections to the vector of erroneous observations $\underline{L}$ and to estimate the vector of unknowns such that the system of equations (3) holds and the square sum $\underline{v} \cdot \underline{v}$ of the corrections is minimized.

In our case, there are two unknowns, m and b, which define the mirror axis in the form $y = mx + b$. The vector of observations contains the coordinates of the vertices, which means that it has 4κ elements.

For each pair of corresponding vertices v_1 (with coordinates x_1 and y_1) and v_2 (with coordinates x_2 and y_2), we introduce two constraints.

The first constraint means that the midpoint between v_1 and v_2 lies on the mirror axis:

$$(y_1 + y_2)/2 = m\,(x_1 + x_2)/2 + b \tag{4}$$

The second constraint means that the vector $\overrightarrow{v_1 v_2}$ is perpendicular to the mirror axis:

$$(x_2 - x_1) + (y_2 - y_1)m = 0 \tag{5}$$

In order to estimate the corrections and unknowns we linearize equations (4) and (5) and apply the common iterative adjustment procedure (Niemeier 2002). In each iteration, we update the unknowns m and b. The initial mirror axis is defined based on the two corresponding vertices with the maximum distance. In addition to the estimates for m and b we obtain a standard deviation s based on the corrections $\underline{v}$. A mirror axis is selected if s does not exceed a user-specified value $s_{max} \in \mathbb{R}_0^+$.

4 Experimental Results

The presented algorithms were implemented in C++ and tested for a data set of 5134 building footprints of the metropolitan Boston area. The data set is freely available as part of the Massachusetts Geographic Information System, MassGIS[1]. According to the data specifications, the building footprints were manually extracted from LiDAR data.

In order to remove marginal details that would hinder the matching process, the building footprints were automatically generalized with an error tolerance ε of 1 m. A neighborhood for the polygons was defined based on a triangulation of the free space not covered by polygons. This triangulation was obtained by using the CGAL[2] library for computational geometry. Each two polygons that were connected with a triangle edge are defined as neighbors.

For each single building and for each pair of neighboring buildings, the type-1 and type-2 matches were searched. Furthermore, for each type-2 match, five iterations of the least-squares adjustment were applied. Together, these computations took 8 seconds on a Windows PC with 3 GB RAM and a 3.00 GHz Intel dual-core CPU.

Table 1. Parameters used for the presented experiments

parameter name	symbol	value
tolerance for building simplification	ε	1 m
tolerance for differences of angles	$\Delta\alpha_{max}$	0.15 rad (≈ 8.6°)
tolerance for differences of edge lengths	Δl_{max}	30%
minimum cardinality for matches	k_{min}	8
maximum standard dev. for mirror axes	s_{max}	1 m

Table 1 summarizes the parameters applied, which were found by experiments. Note, however, that setting $k_{min} = 8$ implies that two symmetry axes are found for a rectangle. Setting k_{min} to a higher value implies that no symmetry axes are found for a rectangle. Therefore, $k_{min} = 8$ is, in a way, a natural choice. The sequence of 90° and 270° turns of building outlines is often very characteristic, thus the tolerance for edge lengths is set to a relatively large value (30%) and to the relatively small value of 8.6° for angles (which can be interpreted as roughly 10% of a right angle).

According to the defined criteria, 11528 type-1 matches and 14100 type-2 matches were found. This means that, for each building, 2.2 type-1

[1] http://www.mass.gov/mgis/lidarbuildingfp2d.htm (accessed 21-10-10)
[2] http://www.cgal.org/ (accessed 21-10-10)

matches and 2.7 type-2 matches were found on average. It is interesting that the type-2 matches are more frequent than the type-1 matches, as it shows that a reflection is indeed a preferred concept in building design (compared to pure repetition). Using the approach based on least-squares adjustment, 10477 mirror axes were found, that is, on average, 2.0 for each building. This also implies that 74% of the type-2 matches indeed represent (pure) axial reflections. We now discuss some selected samples from the data set.

Figure 5 (left) illustrates all type-2 matches found for a set of five apartment buildings. For each match, the corresponding building parts are shown as bold lines. Note that the same part may be involved in multiple matches. Additionally, the figure shows the hypotheses for mirror axes (thin lines). Obviously, many hypotheses are wrong, that is, a translation and/or a rotation need to be performed in addition to the axial reflection in order to match the two shapes. Figure 5 (right), however, shows that correct mirror axes are found by filtering the matches based on the standard deviation that we obtained by least-squares adjustment. Additionally, the least-squares adjustment yields accurate axes. The adjustment process is visualized in Figure 6 for two buildings. In these examples, the initial axes are very inaccurate, but after five iterations we obtain results that are good enough, for example, for visualization.

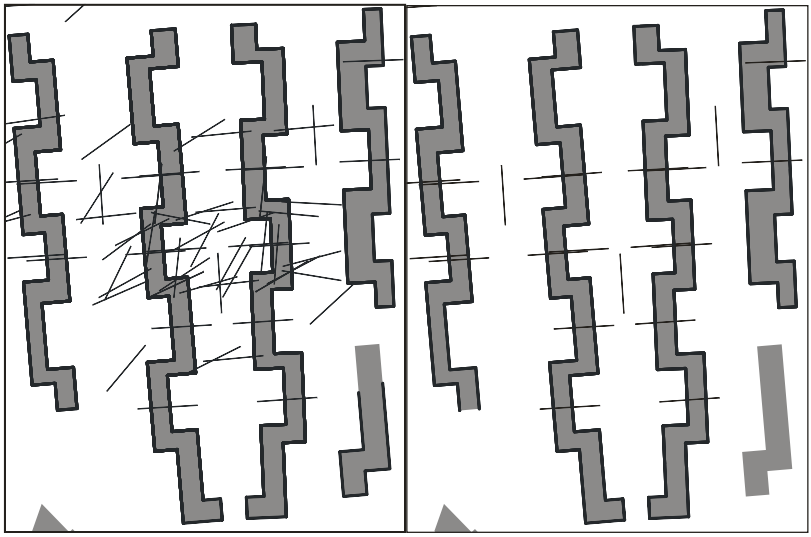


Fig. 5. Hypotheses for mirror axes (left) and selected mirror axes after adjustment (right). The selection of the axes is based on the variance of coordinates that is estimated based on residuals at the polygon vertices. Bold parts of the polygon outlines correspond by symmetry according to the mirror axes shown.

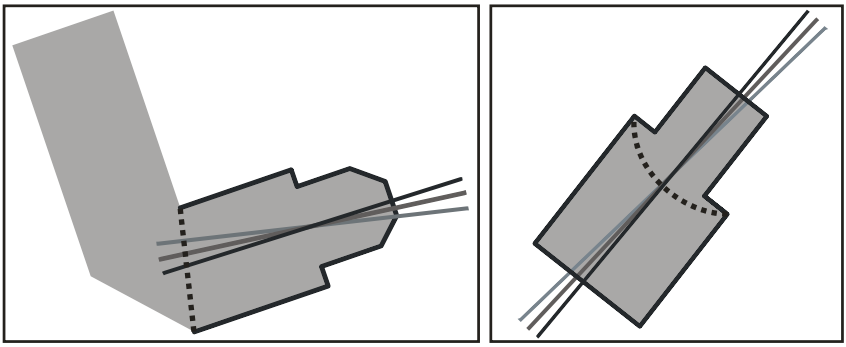


Fig. 6. Illustration of the adjustment process for two buildings with symmetry relations. The figures show the building parts that correspond by symmetry (bold parts of the polygon outlines), the initial mirror axes (light grey lines) the mirror axes after one iteration of the adjustment process (dark grey lines) and after five iterations (black lines). The dashed lines show which pairs of polygon vertices were used to compute the initial axes.

Figures 7, 8, and 9 show the mirror axes that were detected for the samples in Figures 1, 2, and 3, respectively. Generally, the results are satisfactory, that is, the most obvious symmetry relations were found. There are, however, a few open problems that we discuss for the result in Figure 9.

In some cases, we would probably like to report a symmetry relation though each continuous part of the building outline in the relation is small. For example, the mirror axis in Figure 9 labeled with (1) is noticeable but not detected by the algorithm. In this example, there are two continuous building parts that contribute to the symmetry relation, the front façade and the back façade of a building. Since each of the two parts is small (that is, the corresponding string contains less than 8 symbols), the mirror axis is not detected. Together, however, both parts would have the required size. The aggregation of small matches is a problem that still needs to be solved.

Furthermore, the approach based on string matching relies on pair-wise correspondences of polygon vertices or edges. This is problematic, since two shapes can be similar without having such correspondences. We tried to ease this restriction by applying an algorithm for building simplification that removes potentially disturbing details. The problem, however, still occurs in some cases, especially, if the buildings have curved outlines. The mirror axis in Figure 9 labeled with (2) corresponds to a symmetry relation of two buildings with circular arcs. The arcs of both buildings were digitized in two very different ways, thus no vertex or edge correspondences were found. This problem could be solved by detecting arcs in the building outline. For buildings that have a rectilinear shape, however, the algorithm yields good results.

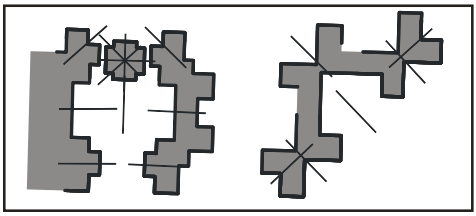


Fig. 7. Detected mirror axes (thin lines) and corresponding building parts (bold lines) for the sample in Figure 1.

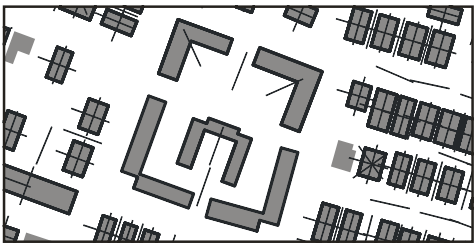


Fig. 8. Detected mirror axes (thin lines) and corresponding building parts (bold lines) for the sample in Figure 2.

Fig. 9. Detected mirror axes (thin continuous lines) and corresponding building parts (bold lines) for the sample in Figure 3. The dashed lines labeled with (1) and (2) display axes that were not detected.

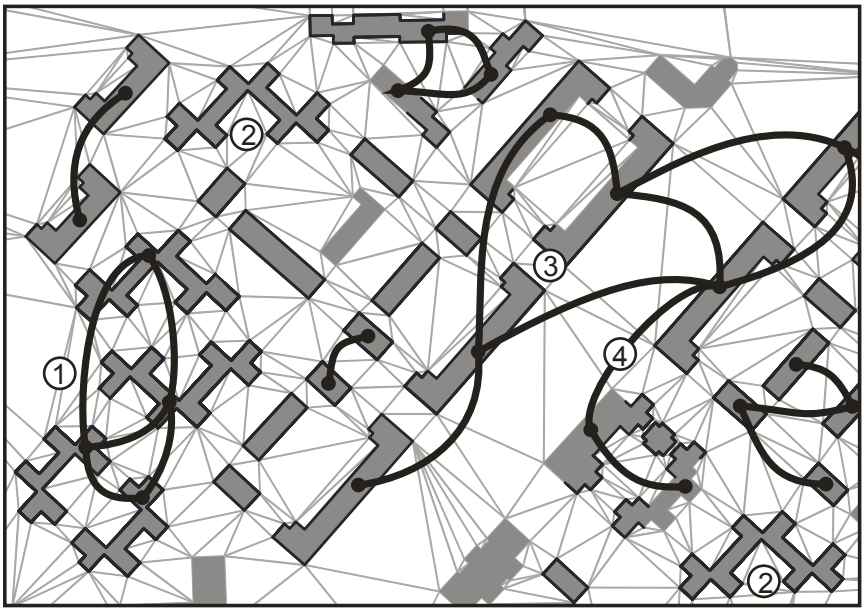


Fig. 10. Detected repetitions (type-1 matches) in building polygons. The bold polygon parts were matched with some other part. The bold arcs link polygons whose parts were matched. The grey lines show edges of the triangulation that was computed to define the neighborhood relation for the buildings. The numbers are referred to in the text.

Finally, we discuss the type-1 matches (that is, repetitions of building parts) yielded by the string-matching method. If we aim to group the buildings according to their similarity, we may be interested in the graph $G(V,E)$ where V is the set of buildings and E contains an edge for each pair of buildings for which at least one type-1 match was found. This graph is illustrated in Fig. 10 (bold arcs). We observe that the connected components of G define a grouping where each group indeed contains buildings of a similar design. For example, the group of four buildings labeled with (1) contains buildings of two different designs that are similar. We find buildings of the same design in different parts of the data set, for example, the buildings labeled with (2). These buildings are not matched because they do not have a similar neighbor. This reflects the proximity criterion in perceptional grouping. Occasionally, we fail to find repetitions (3) or we find matches between buildings that are relatively dissimilar (4). Therefore, additional research on similarity-based grouping is needed. For example, we need to decide how to consider both axial symmetries and repetitions for grouping.

5 Conclusion and Outlook

We have discussed the problem of finding symmetry relations and mirror axes in geospatial datasets of buildings. This problem is important for the solution of map generalization problems, landmark detection, and building classification. The presented algorithm for symmetry detection uses a very efficient string-matching approach based on dynamic programming. Mirror axes are found using an approach based on least-squares adjustment. The algorithm copes both with geometric errors and partial symmetries.

The results that we discussed in this paper show that the proposed method allows us to process large datasets fast (that is, several thousands of buildings in a few seconds) and to find most of the dominant symmetry relations. On average, for each building two mirror axes were found. In addition to the symmetry axes, the algorithm yields matches of similar building parts.

Future work is needed to aggregate symmetry relations. This is important, since symmetry relations involving multiple disconnected building parts are currently not considered in the algorithm proposed.

Furthermore, it is planned to integrate the derived information into methods for map generalization. We can expect that using the information derived with the presented algorithm will clearly improve the results of map generalization, in particular, building simplification and aggregation.

References

Atallah, M. J. (1985). On Symmetry Detection. IEEE Transactions on Computers, c-34(7), 663–666.

Cho, M. and Lee, K. M (2009). Bilateral Symmetry Detection via Symmetry-Growing. In: Proc. British Machine Vision Conference (BMVC '09).

Christophe, S. and Ruas, A. (2002). Detecting Building Alignments for Generalisation Purposes. In: Proc. ISPRS Commission IV Symposium on Geospatial Theory, Processing and Applications. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXIV, part 4.

Gaffuri, J. and Trévisan, J. (2004). Role of Urban Patterns for Building Generalisation: An Application of AGENT. In: Proc. 7th ICA Workshop on Generalisation and Multiple Representation.

Gusfield, D. (1997). Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press.

Haunert, J.-H. and Wolff, A. (2010). Optimal and Topologically Safe Simplification of Building Footprints. Pages 192–201 of: Proc. 18th ACM

SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM-GIS'10).

Heinzle, F. and Anders, K.-H. (2007). Characterising Space via Pattern Recognition Techniques: Identifying Patterns in Road Networks. Chap. 12, pages 233–254 of: Mackaness, W., Ruas, A., and Sarjakoski, T. L. (eds), Generalisation of geographic information: Cartographic modelling and applications. Elsevier.

Knuth, D. E., J. H. Morris, Jr., and Pratt, V. R. (1977). Fast Pattern Matching in Strings. Siam Journal on Computing, 6(2), 323–350.

Lladós, J., Bunke, H., and Martí, E. (1997). Using Cyclic String Matching to Find Rotational and Reflectional Symmetries in Shapes. Pages 164–179 of: Intelligent Robots: Sensing, Modeling and Planning. Series in Machine Perception and Artificial Intelligence, vol. 27. World Scientific.

Loy, G. and Eklundh, J.-O. (2006). Detecting Symmetry and Symmetric Constellations of Features. Pages 508–521 of: Proc. 9th European Conference on Computer Vision (ECCV '06), Part II. Lecture Notes in Computer Science, vol. 3952. Springer.

Mackaness, W. and Edwards, G. (2002). The Importance of Modeling Pattern and Structure in Automated Map Generalisation. In: Proc. Joint ISPRS/ICA Workshop on Multi-Scale Representations of Spatial Data.

Mitra, N. J., Guibas, L. J., and Pauly, M. (2006). Partial and Approximate Symmetry Detection for 3D Geometry. ACM Transactions on Graphics, 25(3), 560–568.

Neun, M., Burghardt, D., and Weibel, R. (2008). Web Service Approaches for Providing Enriched Data Structures to Generalisation Operators. International Journal of Geographic Information Science, 22(2), 133–165.

Niemeier, W. (2002). Ausgleichungsrechnung. Walter de Gruyter.

Park, M., Lee, S., Chen, P.-C., Kashyap, S., Butt, A. A., and Liu, Y. (2008). Performance Evaluation of State-of-the-Art Discrete Symmetry Detection Algorithms. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08).

Peters, D., Wu, Y. H., and Winter, S. (2010). Testing Landmark Identification Theories in Virtual Environments. Pages 54–69 of: Spatial Cognition. Lecture Notes in Computer Science, vol. 6222. Springer.

Regnauld, N. (2003). Algorithms for the Amalgamation of Topographic Data. In: Proc. 21st International Cartographic Conference (ICC '03).

Ruas, A. and Holzapfel, F. (2003). Automatic Characterization of Building Alignments by Means of Expert Knowledge. In: Proc. 21st International Cartographic Conference (ICC '03).

Steiniger, S. (2007). Enabling Pattern-Aware Automated Map Generalization. PhD thesis, University of Zürich.

Steiniger, S., Burghardt, D., Lange, T., and Weibel, R. (2008). An Approach for the Classification of Urban Building Structures Based on Discriminant Analysis Techniques. Transactions in GIS, 12(1), 31–59.

Thomson, R. C. and Brooks, R. (2002). Exploiting Perceptual Grouping for Map Analysis, Understanding and Generalization: The Case of Road and River

Networks. Pages 148–157 of: Proc. 4th International Workshop on Graphics Recognition Algorithms and Applications. Lecture Notes in Computer Science, vol. 2390. Springer.

Werder, S., Kieler, B., and Sester, M. (2010). Semi-Automatic Interpretation of Buildings and Settlement Areas in User-Generated Spatial Data. In: Proc. 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS '10). To Appear.

Wertheimer, M. (1938). Laws of Organization in Percetional Forms. Pages 71–88 of: A Source Book of Gestalt Psychology. Routledge & Kegan Paul.

Wolter, J. D., Woo, T. C., and Volz, R. A. (1985). Optimal Algorithms for Symmetry Detection in two and three Dimensions. The Visual Computer, 1(1), 37–48.

Yan, H., Weibel, R., and Yang, B. (2008). A Multi-Parameter Approach to Automated Building Grouping and Generalization. GeoInformatica, 12(1), 73–89.

Yang, X., Adluru, N., Latecki, L. J., Bai, X., and Pizlo, Z. (2008). Symmetry of Shapes Via Self-Similarity. Pages 561–570 of: Proc. 4th International Symposium on Advances in Visual Computing, Part II. Lecture Notes In Computer Science, vol. 5359. Springer.

Simultaneous & topologically-safe line simplification for a variable-scale planar partition

Martijn Meijers

Delft University of Technology (OTB – Department of GIS Technology)
Delft, the Netherlands
b.m.meijers@tudelft.nl

Abstract. We employ a batch generalization process for obtaining a variable-scale planar partition. We describe an algorithm to simplify the boundary lines after a map generalization operation (either a merge or a split operation) has been applied on a polygonal area and its neighbours. The simplification is performed simultaneously on the resulting boundaries of the new polygonal areas that replace the areas that were processed. As the simplification strategy has to keep the planar partition valid, we define what we consider to be a valid planar partition (among other requirements, no zero-sized areas and no unwanted intersections in the boundary polylines). Furthermore, we analyse the effects of the line simplification for the content of the data structures in which the planar partition is stored.

1 Introduction

A planar partition is a tessellation of the 2D plane into polygonal areas. These polygonal areas form a complete covering of the domain without overlaps and gaps. To obtain a variable-scale planar partition stored in the topological Generalized Area Partition (tGAP) data structures, we employ an off-line map generalization process as a pre-processing step (before online use, see for more details Van Oosterom 2005; Meijers et al. 2009). The boundaries of the partition are stored in the form of polylines and topological references. The tGAP structure can be used in a networked context (e.g. the Internet) to supply a vector map at a variety of map scales

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_17,

(many more than usually stored in a Multi-Resolution Database (MRDB)). Initially, for storing the geometry of the boundaries, the use of a forest of Binary Line Generalisation (BLG) trees was proposed (Van Oosterom 2005). A BLG tree (Van Oosterom 1990) is a binary tree and stores the result of the Douglas-Peucker line simplification algorithm. Advantages of employing the forest of BLG trees would be that the data structure would contain as little geometric redundancy as possible. However, the Douglas-Peucker algorithm does not give any guarantees on topological correctness and we noticed that the use of the BLG trees would create communication overhead in a situation where a client application retrieves the map data from a server when the trees are only partially transmitted to obtain the right amount of vertices in the polylines.

An alternative we investigated was not to simplify the boundaries at all, but to keep the original geometry of the boundaries with which we started. We quickly noticed that during use of the resulting variable-scale planar partition in a network context, the number of vertices in the boundaries was too high, especially for the smaller map scales, leading to a slow performing user interface.

Therefore, we turned back to simplify the boundaries, but now store the result of the simplification explicitly (thus *not* deriving the geometry dynamically from a special reactive data structure, like with the BLG trees, as this leads to administrative overhead) and allow some redundancy in the data structure (but preferably as minimal as possible). The line simplification has to be performed without violating any of the requirements for a valid variable-scale planar partition. In this paper, we give an overview of how we perform the simplification on a *subset* of the boundary polylines in the planar partition (the resulting lines from a higher level generalization operation, such as aggregation or splitting a polygonal area over its neighbours). The removal of points leads to short cuts in the polylines. It is ensured that taking such a short cut does not lead to topological problems (the boundaries are not allowed to change their relative position). However, we also observed that a spatial configuration that leads to a problem (e.g. a change of side or occurrence of an intersection) at first might be changed later, because another point has been removed. Our approach also deals with these issues. The simplification is stopped when a certain criterion is reached (enough points have been removed or there are no more points that can be removed without violating the requirements for a valid planar partition).

The research questions that we try to answer are:

- How to prevent topological errors and what are sufficient conditions to guarantee topological correctness in a variable-scale environment Plümer and Gröger 1997)?

- Which lines do we simplify after applying a generalization operator on the polygonal areas?
- When to stop the simplification?
- What are the effects on the contents of the data structures when applying the line simplification?

The remainder of the paper is structured as follows. We review related work in Section 2. In Section 3, we formalize the requirements for a variable-scale planar partition. In Section 4, we improve a known method for topologically safe simplification for more than one polyline as input (cf. Kulik et al. 2005). The input polylines will be simplified simultaneously – thus not one after the other. Our improvements focus on an efficient implementation, keeping the polygonal areas explicitly valid (note that the areas can contain holes in their interior) and the tGAP structure as context. Furthermore, we use a step-wise generalization process in which we only simplify a subset of the boundaries, thus not all polylines will be simplified at the same time. Section 5 shows how we tested and analysed the algorithm. Section 6 concludes the work and gives some suggestions for future work.

2 Related work

In literature, a multiplicity of methods is known to simplify (cartographic) lines. Saalfeld (1999) gives a classification of polyline simplification methods:

in vacuo modifies one polyline in isolation, possibly leading to topological conflicts that have to be resolved by post-processing;

en suite modifies a single polyline in context (looking at topological relationships with nearby features); and

en mass modifies the complete collection of polylines and all other features of a map, taking the topological relationships into consideration during adjustment.

Apart from the classification given by Saalfeld, the algorithms can be divided in two main groups: using *refinement* (i.e. an approach from coarse to fine, starting with a minimal approximation of a polyline and then adding the most significant points, until a prescribed tolerance is met) or using *decimation* (i.e. an approach which starts with the most detailed version of a polyline and then eliminates the least important points first, thus going from fine to coarse).

The most known algorithm for simplifying lines, *in vacuo* using a refinement approach, is the Douglas-Peucker line simplification (Ramer

1972; Douglas and Peucker 1973). It was modified by Saalfeld (1999) to work on a polyline *en suite*. Da Silva and Wu (2006) argued that topological errors could still occur and gave an extension to the suggested approach. However, their approach is not explicitly designed for keeping a planar partition valid as they cannot ensure that polygonal areas keep size.

Another *en suite* algorithm is developed by De Berg et al. (1998). The core of the algorithm is also used for simplifying polylines in a planar subdivision (*en mass*), but each polyline in the main loop of their algorithm is still simplified *en suite* (so the simplification outcome depends on the order of processing the polygonal chains).

A better approach in this respect is the one given by Kulik et al. (2005), which simplifies the polylines simultaneously (thus not one after the other). The basis for their recipe is the algorithm described by Visvalingam and Whyatt (1993). It is using decimation for simplifying lines *in vacuo*. The algorithm of Visvalingam and Whyatt was extended by Barkowsky et al. (2000) using different criteria for the order in which points will be removed (leading to different shapes as output). Kulik et al. (2005) developed the approach for simplifying polylines *en mass*, but they consider only a connected graph for the topology aware simplification (the algorithm in this paper also deals with an unconnected graph, in case of islands in the polygonal areas, e.g. face 4 in figure 1). Furthermore, in their description of the algorithm they show that it is necessary to check after every simplification step whether points that could not be removed before are now allowed to be simplified. It appears that their algorithm in this case can lead to quadratic running times. Also, it is not clear in their description how near points that might influence the simplification can be obtained efficiently in an implementation.

Dyken et al. (2009) also present a method for simultaneous simplification of a collection of polylines in the plane (simplifying them *en mass*). The method is based on building a triangulation. Although this approach seems promising, building a triangulation after every generalization operation will be expensive from a computational point of view, mainly because we already have a topological graph at hand.

It must be noted that none of the methods described above discuss line generalization in a stepwise generalization process, thus intermingled with other generalization operations, such as merging and splitting of polygonal areas (aggregation) in a planar partition for a variable-scale context.

3 A valid planar partition at a fixed and at a variable map scale

A planar partition can be fully described by storing a topological structure. Polylines form the boundaries of the polygonal areas; each polyline has two endpoints, which we call the *exterior* points, and the rest of the points of the polyline are called *interior* points. Following Gröger and Plümer (1997), we give a set of requirements for this boundary representation, so that the resulting model is a *valid* planar partition of polygonal areas. The mathematical concept of a graph G(N,E) consists of a set of nodes N and a set of edges E, where the edges define a relationship between the nodes. Let us consider the set of instances $n \in N$ and $e \in E$. If a relationship between a node n_0 and an edge e_0 exists, the two instances are called *incident*. The *degree* of a node is the number of its incident edges. If a node n_1 is also incident to an edge e_0, the nodes n_0 and n_1 are said to be *adjacent*. Using the graph concept, we can specify a set of requirements for the boundaries (as illustrated in figure 1):

1. With a graph $G(N,E)$, we model the geometric relationship of the endpoints of the polylines: when two endpoints have the exact same coordinates, they become a node in the graph; thus N is the set of geometrically unique endpoints and E the set of polylines. We embed G in the 2D plane by the location of the points of the polylines and we specify that G is a planar drawing. This implies that polylines are only allowed to touch at their endpoints, no intersections or overlaps are present and each polyline must be simple (also no self-intersections are allowed).
2. All nodes in G must have a *degree*>1. This prevents having dangling polylines as a boundary.
3. As a result of the fixed embedding of the graph, we can define each face f of G as the maximal connected region of the 2D plane where every two points x and y in f can be connected to each other without intersecting or touching any edge $e \in G$. The edges that delimit f form its boundary b. The edges $e \in b$ form one (or more) cycle(s). For each cycle there exists a path $n_0, e_0, n_1, e_1, ..., n_{i-1}$, in which endpoints and polylines alternate and where endpoint $n_0 = n_i$.
4. Each face is delimited by at least 1 cycle (holes in the interior of a face are thus allowed). If a face has more than 1 cycle, these cycles have to be nested properly geometrically (if this is the case, one of these cycles should contain all other cycles). This nesting can only be

1 level deep on account of the previous requirement ('connected interior').

5. Each polygonal area in the planar partition corresponds to exactly one face (thus no multi-part polygonal areas are allowed) and each face corresponds to exactly one polygonal area. This symmetry enforces that all polygonal areas form a complete covering of the domain, without overlaps.
6. In *G*, there exists one unbounded face (universe), which has exactly one cycle (geometrically containing the cycles of all other faces). Furthermore, the set of faces *f* is *F*.

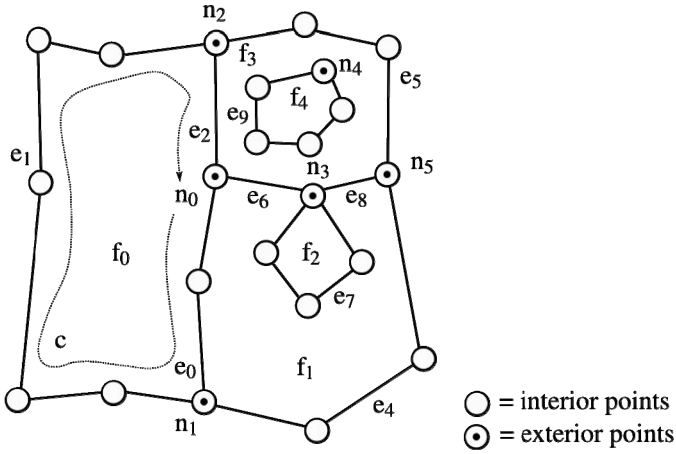


Fig. 1: Face f_0 is delimited by the boundary cycle *c* (i.e. the path $n_0, e_0, n_1, e_1, n_2, e_2, n_0$). The boundary cycle that delimits f_1 ($n_0, e_6, n_3, e_7, n_3, e_8, n_5, e_4, n_1, e_0, n_0$) is not simple (it passes through n_3 twice). However, f_1 forms 1 maximal connected region. Face f_3 is delimited by two cycles (one starting at n_5 and one starting at n_4), which are properly nested. Node n_4 has a *degree*=2.

From *G*(*N*,*E*) we can derive its dual *G**(*F'*,*E'*). This dual graph *G** models the *adjacency* relationships of the polygonal areas, i.e. in this graph *G** the faces *F'* form the nodes and the edges *E*' are the polylines one has to cross for neighbouring faces $f \in F'$.

So far, we were only concerned about the planar partition at one fixed map scale. We now extend the approach to a tessellation of the 2D plane at a variable (i.e. not pre-defined) map scale. For this, we need generalized

versions of the planar partition P. These versions with a lower level of detail can be seen as a third dimension. To obtain all versions, we iteratively apply an operation O on the planar partition P, which selects a polygonal area that should be generalized and modifies P accordingly, outputting P'. Symbolically, we can describe this as: $O(P) \rightarrow P'$. The generalization operator must ensure that the output it delivers (P') is again a valid planar partition that fulfils all requirements given above, e.g. plain removal of a polygonal area is thus not allowed, as this would violate the requirement of complete coverage of the domain (and would create a gap).

We allow two types of generalization operators to modify the number of faces in P: for a merge operation, we remove the boundary polyline(s) between a candidate and its most compatible neighbour and form a new boundary cycle for this neighbour; and for a split operation, we remove all boundaries associated with the polygonal area that is split, we introduce new boundaries between the neighbours of this area, and we form new boundary cycles for these neighbours. With both operations the dual graph G^* can be used to find the neighbours.

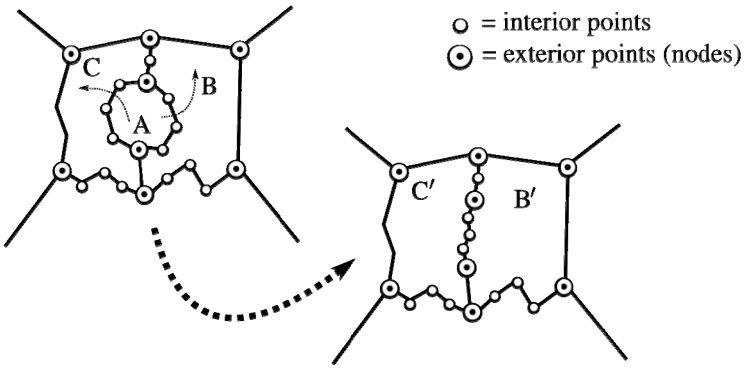
Based on the observations above, to have a variable-scale tessellation of the 2D plane, we add two more requirements to our list:

1. Every generalization operator O applied to P must output a *valid* planar partition P'.
2. Hierarchically speaking the new polygonal areas and boundaries from P' must be adjacent to and must not overlap with the remaining and unchanged areas and boundaries from P (in the map scale dimension).

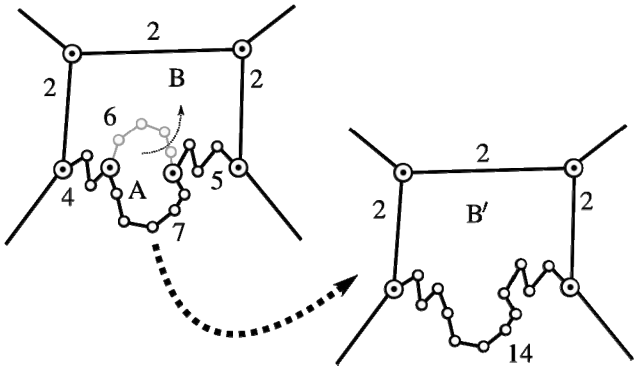
4 Line simplification

4.1 The need for line simplification

We use a stepwise map generalization process. This process records all states of the planar partition after applying a generalization operator in a tree structure. With the obtained hierarchy the average number of polygonal areas shown on a user's screen can be kept roughly equal, independent from the size of the user's view port (by varying the level of detail when a user zooms in or out, thus showing objects closer to or further from the top of the hierarchical structure). The removal of area objects (by merging or splitting them) leads to less objects per region. A user zooming out leads to an enlarged view port and ascending the hierarchy can supply an equal number of more generalized (thus larger) area objects to an end user, similar to the number before the zoom action.



(a) Splitting of polygonal areas leads to unwanted nodes



(b) Unwanted nodes also result from merging two polygonal areas. Furthermore, the average number of coordinates per boundary increases

Fig 2. Both (a) and (b) show that unwanted nodes can exist after a split or a merge operation. Furthermore, it is illustrated that not simplifying merged edges leads to an increased average number of coordinates per boundary.

However, the related boundaries of the polygonal objects will get more coordinates (per object) if the original boundary geometry is kept and not simplified. As can be observed in Figure 2, a split operation, e.g. implemented using triangulation, like in Bader and Weibel (1997), can lead to unnecessary nodes in the topology graph (nodes where degree = 2). This also happens when a merge operation is performed (see Figure 2(b)). Therefore, we merge the boundaries that are incident to those nodes. However, this merging leads to boundaries with more coordinates.

The increase in the number of coordinates is illustrated by the example shown in Figure 2(b). Polygonal areas A and B are merged. This leads to

two nodes with *degree*=2. On average, the number of coordinates before the area merge operation in the boundaries is (2+2+2+4+7+5+6)/7=28/7=4. After the merge, we can remove the two *degree*=2 nodes and thus merge the boundaries which leads to: 4+7+5–2=14 coordinates for this new boundary. On average the number of coordinates of all the boundaries is: (14+2+2+2)/4=20/4=5, which is more than before the merge operation. According to our rule that we want to keep the number of vertices per polyline equal, the polylines have to be simplified.

4.2 An overview of the simplification procedure

We employ a decimation approach for simplifying the selected boundary polylines. The order of removing points is determined by a weight value *w*, which we calculate for each interior point of the polylines to be simplified. For calculating the weight values, we get 3 consecutive points, p_{i-1}, p_i, p_{i+1} from a polyline forming a triangle τ. In our implementation the weight is calculated based on the area of the associated triangle τ, i.e. $\Box(p_{i-1}, p_i, p_{i+1})$, and therefore completely based on geometry (cf. Visvalingam and Whyatt 1993). There could be more geometrical criteria, like sharpness of turn angle, length of sides, ratio of sides, etcetera (alternatives are discussed in Barkowsky et al. 2000). Note that Kulik et al. (2005) also assign a 'semantic' weight per point (next to the 'geometric' weight), which they base on the feature class of the polyline, where the point belongs to and is also dependent on the user's application domain.

The exterior points of the polylines (forming a node in the planar partition) cannot be removed. At each step, the interior point p_i having the overall lowest weight value will be removed, leading to a 'collapse' of triangle τ into a *short cut* $\overline{p_{i-1,} p_{i+1}}$. Our simplification strategy has to obey the requirements that we have given for a planar partition, thus not all short cuts will be allowed. We observed that a spatial configuration that leads to a problem at first might be changed later, because another point has been removed (that was preventing a collapse). The algorithm re-tries removal of the blocked point in this case.

4.3 Dynamic and static polylines and how to select those

Two types of polylines that play a role in the simplification can be distinguished: *dynamic* polylines that will be simplified, i.e. interior points can be removed as long as no intersections or degeneracies in the planar parti-

tion requirements are caused by this removal; and *static* polylines that will not be simplified and for which all points are fixed (these points can forbid certain short cuts in lines that are simplified). Points of the first type are termed dynamic points and points of the second type are termed static points. Points that eventually will be removed by the simplification algorithm have to be interior and dynamic points.

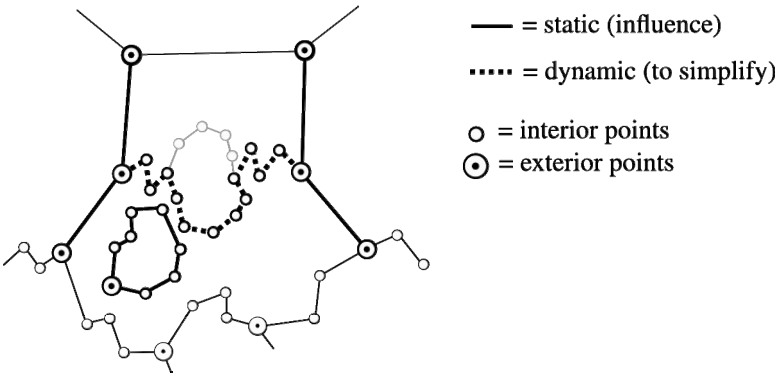


Fig. 3. Dynamic polylines will be simplified (only one in this figure), static polylines can have an influence on the simplification. Note that the alternative is illustrated in which only the polylines that are incident to a merge boundary will be simplified.

After a merge or split generalization operation is finished we must chose which lines to simplify (thus select the *dynamic* polylines). Two viable alternative approaches are:

1. Simplify the polylines that are (in case of an area merge operation) incident to the common merge boundaries or (in case of an area split operation) simplify the new boundaries that stem from the split operation; and
2. Simplify all polylines of the resulting new area(s).

As the simplification should be topology-aware, the *static* polylines in the neighbourhood also have to be selected as input for our algorithm as these can influence the outcome of the simplification. For this purpose, we can use the topology structure to select the lines that are in the vicinity of the lines that we want to simplify. We use the topology structure as an initial spatial filter (going from neighbouring areas to their related boundaries); then with a second pass we can select the related boundaries based on bounding box overlap with the union of the bounding box of all dynamic polylines. An alternative approach is to keep an auxiliary data structure (such as an R-tree or quad-tree) for fast selection of the polylines in

the vicinity. Downside of this approach is that an auxiliary structure needs to be kept, while the topology structure is already present. However, the initial filtering step using the topology structure can be expensive if the new polygonal area is at the border of the domain (leading to a selection of *all* edges at the border of the domain that have to be filtered on their bounding box).

4.4 A stop criterion for the simplification

We iteratively remove points from all dynamic input polylines, until a certain optimal goal is reached. We have two main choices for defining this optimal goal (to stop the simplification):

eps-stop Use a geometric measure as a threshold ε (all points having their weight $w<\varepsilon$ should be removed, where w is based on the size of the triangle of 3 consecutive points in the polyline).

Using this approach, we could use a fixed ε throughout the whole process of building the variable-scale hierarchy. This is not a very realistic option as the number of polygonal areas (and thus the level of detail) decreases when more generalization operators have been applied (when more polygonal areas have been merged or split, the remaining boundaries should also be simplified more). A better option is to determine dynamically the value of ε with every generalization step. For this we can:

- take the average or median value of all weight values as ε (all points having a weight value smaller than this have to be removed);
- set an ε based on other criteria, like the smallest segment length of all polylines taking part in the simplification. Such an alternative choice for ε also means that the weight values w for all interior points have to be calculated accordingly.

count-stop Use a fixed number of points that we want to see removed.

Using a fixed number of output points as the optimal goal, we can count the number of points in the input and try to remove a certain percentage. Two similar, but somewhat different options in this respect are:

- take a local approach: e.g. per input polyline try to remove half of the points (but do not remove more points from a polyline than half of its original points); or
- take a regional approach: for all polylines being simplified, count the total number of points and keep removing points, until in total half of these points have been removed.

Note that both approaches can leave more points as a result than wished for, because some of the points can be blocked by others (because topo-

logical errors must be prevented), although they fulfil the condition for removal (e.g. $w<\varepsilon$). Note also that with both approaches we can vary the percentage of points that we want to remove (instead of half of the points) depending on how far we want to 'push' the generalization. In an extreme case, we could set the percentage to such a value that the algorithm will try to remove all points leading to straight lines as much as possible (only topological 'problematic' points are remaining).

4.5 To prevent topological errors

The algorithm

An outline of the procedure is depicted in Algorithm 1. For all dynamic polylines, a doubly-linked list is created (storing the points in the order in which they are present in the original polyline, cf. Algorithm 1, line 1). Further, for all interior points of these polylines, a weight w is calculated. Important points get a higher weight than less important ones.

All dynamic and interior points are inserted in a priority queue Q, ordered by their weight values w (Algorithm 1, line 3). In our implementation we use a red-black tree (Guibas and Sedgewick 1978) for the priority queue. Points with equal weights are dealt with in the order of insertion. In-order traversal of the red-black tree Q allows now to find the point with the smallest weight value, which is then removed from Q. For point p_i, its neighbours, p_{i-1} and p_{i+1} can be retrieved from the polyline doubly-linked list. The three points together form the triangle τ (see Figure 4).

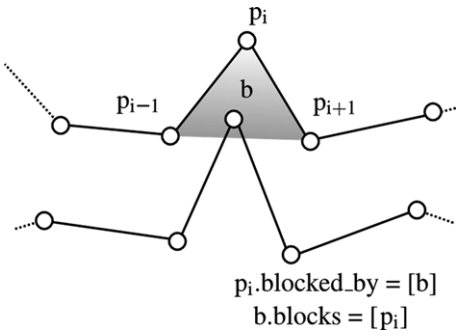


Fig. 4. As b blocks the removal of p_i, the blocks and blocked by lists are filled accordingly.

Algorithm 1 Simplification, while keeping a planar partition valid
Input: A set of dynamic polylines and a set of static polylines
Output: A set of simplified polylines

```
  {pre-processing}
1: Create doubly-linked list for each dynamic polyline
2: Compute weights w for all interior points of dynamic polylines
3: Add dynamic, interior points to priority queue Q based on weights
4: Create pointers between points of static polylines with only 2 points
5: Create kd-tree of all points of both dynamic and static polylines
  {simplifying}
6: while Q not empty do
7:      Pop least important p_i from Q
  {stop criterion, see section 4.4}
8:      if stop criterion met for p_i then
9:           break
10:     allowed ← True
11:     if p_i part of loop edge with 4 points then
12:          allowed ← False {no more 'tries' for this point}
13:     Retrieve τ (using p_i-1 and p_i+1 from linked list)
14:     vicinity ← search kd-tree for points near p_i using box of τ
15:     for all b ∈ vicinity do
16:          if b ∉ (p_i-1, p_i , p_i+1) and b part of segment p_i−1, p_i+1 then
17:               allowed ← False {no more 'tries' for this point}
18:     if allowed then
19:          for all b ∈ vicinity do
20:               if b ∉ (p_i-1, p_i , p_i+1) and b on τ then
21:                    allowed ← False
22:                    Append b to p_i.blocked_by list
23:                    Append p_i to b.blocks list
24:     if allowed then
25:          Remove p_i from linked list
26:          Adjust weights for p_i-1 and p_i+1
27:          Check whether p_i-1 and p_i+1 are still blocked, otherwise add to Q
28:          Mark p_i as removed in kd-tree
29:          for all u ∈ p_i.blocks do
30:               Remove p_i from u.blocked_by list
31:               if u.blocked by list empty then
32:                    Add u to Q
  {output}
33: return Simplified polylines by traversing doubly-linked lists
```

The short cut that will be taken is $\overline{p_{i-1}, p_{i+1}}$. Such a short cut is only allowed if it does not lead to an *invalid* planar partition, i.e. violates one of the requirements, as described in section 3. Any intersections of the new short cut with other polylines or another segment of the polyline itself (i.e. a line between two consecutive points of the polyline) have to be prevented. As the partition is valid to begin with (which can be ensured by us-

ing a constrained triangulation, see Ledoux and Meijers 2010; Ohori, 2010), the polylines of the planar partition do not contain any (self-) intersections. An intersection of the short cut can only be created when a segment σ 'enters' τ via the open side $\overline{p_{i-1,}p_{i+1}}$ (as it is not allowed to enter or leave the area of τ via either $\overline{p_{i-1,}p_i}$ or $\overline{p_{i,}p_{i+1}}$; this immediately would lead to an intersection). A point of σ must thus be interacting with τ for an intersection to happen and it is sufficient to check whether such a point exists, to prevent this. Points that can influence the collapse are termed *blockers*. These blockers stem from:

1. the polyline itself (self-intersection); or
2. other polylines in the vicinity of τ (both static and dynamic).

To efficiently find those points, we use a kd-tree (not just a regular kd-tree, but one following Bentley (1990) for which the tree does not need to be re-organised after removal of points, but to which no extra points can be added after initial organisation of the tree). All (interior as well as exterior) points of all polylines taking part in the simplification are inserted in this kd-tree (algorithm 1, line 5). The bounding rectangle around the triangle τ is used to query the kd-tree to find all points in the neighbourhood of this triangle to see if there are any blockers for creating the shortcut $\overline{p_{i-1,}p_{i+1}}$. If a blocker is found, the short cut is not taken and as p_i was removed from Q it will not turn up in the next iteration.

As the kd-tree contains the points of all dynamic polylines, a potential blocker b can be a point that forms the triangle τ. If this happens we do not check whether b blocks p_i (i.e. p_{i-1}, p_{i+1}, nor p_i itself can block removal of p_i) or no simplification could take place at all. Since a blocker b can be removed itself later on and then a short cut for this vertex p_i might be allowed, a cross reference is set up between p_i and b (b is registered in the 'blocked by'-list of p_i and p_i is registered in the 'blocks'-list of b).

If no blockers were found, p_i can be removed from the doubly-linked polyline list it belongs to (creating a short cut in this polyline). The point is also marked as removed in the kd-tree. If the removed point p_i was a blocker itself (having one or more points in its 'blocks' list), it removes itself from the 'blocked by' list of these particular points. If for a point u its 'blocked by' list becomes empty (because of the removal of p_i), u is placed back again in Q, so it has a chance of being a short cut in the next iteration

(if then not blocked by any other point and still not having fulfilled the condition for removal, e.g. having a weight $w<\varepsilon$). If one of the two neighbouring points p_{i-1} or p_{i+1} was blocked, it is also checked whether this is still the case (the shape of their related triangle also has changed, because of the short cut operation).

The algorithm ends when the chosen criterion has been met, i.e. there are no more points that can fulfil the criterion to reach the optimal goal, and the new polylines are returned.

More cases for validity

Apart from intersection prevention by testing near points, more specific situations have to be taken into account, because of the validity requirements of the planar partition. Two other conditions also have to be checked (illustrated in Figure 5) to prevent occurrence of zero-sized polygonal areas:

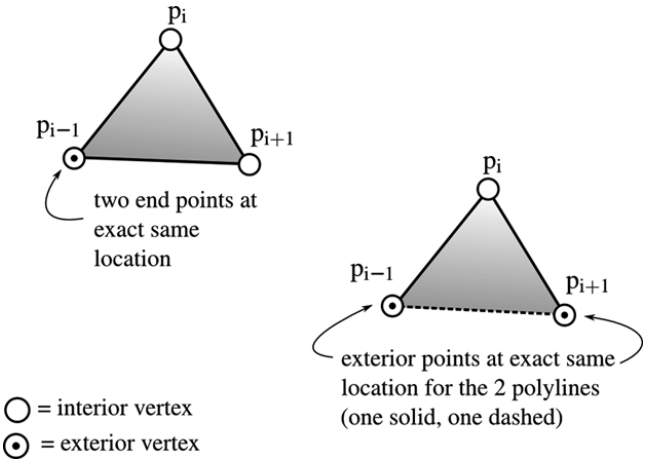


Fig. 5. If taking a short cut leads to a polygonal area that has no area, we put the end points as blockers for p_i. The result is that p_i is not removed, as the endpoints of the polylines will never be removed.

1. Same polyline (see Algorithm 1, line 11): A special check is performed when p_{i-1} or p_{i+1} is the endpoint of a so-called 'loop' polyline (a special case where the 2 exterior points of the polyline are at the exact same location, cf. Figure 5-left). We now have to check whether there will still be enough points in the polyline when we take away p_i (because no zero-size area is allowed). We can do this by travers-

ing the linked list and check when p_{i-1} is a loop endpoint, whether p_{i+2} is also such an endpoint (similar with p_{i-2} for p_{i+1}). Note that this is a rare case (only the two last interior points for a triangular face).

2. Different polyline (see Algorithm 1, line 16): Another check is performed on whether $\overline{p_{i-1}, p_{i+1}}$ is already connected by another polyline (by allowing twice such a polyline, a zero-size area would be created, Figure 5-right). To prevent this, it is necessary to check if a potential blocking point *b* returned by the kd-tree is part of such a polyline between p_{i-1} and p_{i+1}: a. for a dynamic point returned by the kd-tree, it is possible to use the doubly-linked list to navigate to the next vertex and check whether p_{i-1} and p_{i+1} are fixed; and b. for a static point returned by the kd-tree, we put an extra pointer to the other endpoint of the static polyline if the line only consists of two points. This allows checking whether a static point blocks the collapse of τ.

5 Experiments

We implemented the line simplification algorithm in our tGAP test environment. This environment is using the PostgreSQL[1] database system with PostGIS[2] extension. The algorithm is implemented in the Python[3] programming language. With the implementation we tried different alternatives.

Table 1. Symbolic names of the alternatives that were tested.

	Simplify which edges?	
Stop criterion?	Merged edges only at nodes with degree = 2	All edges of new area (including merged ones)
No simplification at all	none	none
Count stop (regional, half # of interior points)	m_ct	ct
Eps stop (median of all weights)	m_eps	eps
As far as possible	m_full	full

In total, we tested 7 alternatives — with only merge operations applied to the polygonal areas — for which the symbolic names are shown in Ta-

[1] www.postgresl.org
[2] postgis.refractions.net
[3] www.python.org

ble 1. The first alternative (labelled 'none' in Table 1) we tested, was merging edges at nodes with *degree*=2, but not applying any simplification. This was meant as a reference test as we already knew that this would lead to too many coordinates per boundary. The remaining strategies come from varying two alternatives: which lines to simplify (only the merged boundaries, prefixed with 'm_' in Table 1, or all boundaries of the new area); and when to stop the generalization (based on the median ε-value for all boundaries being simplified – dynamic eps-stop –based on the number of points – the regional count-stop approach, or simplify as far as possible, respectively labelled 'eps', 'ct' or 'full').

Table 2 shows the number of polylines and their average number of coordinates for the datasets we used in our experiment. We tested with three datasets representing different types of geographic data. We used a topographic, urban dataset; a topographic, rural dataset; and a land use dataset. Both topographic datasets represented infrastructure objects, which were not present in the land use dataset.

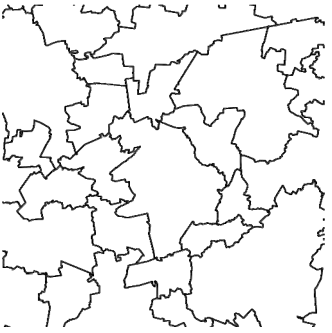
Table 2. For the datasets used in our experiment, the number of polygonal areas, polylines and average number of coordinates per polyline at start.

Dataset	# of areas at start	# of polylines at start	avg # coords per polyline	total # coords
Topographic, urban	9,381	24,528	4.6	112,828
Topographic, rural	3,286	8,212	10.6	87,047
Land use	5,537	16,592	7.2	119,462

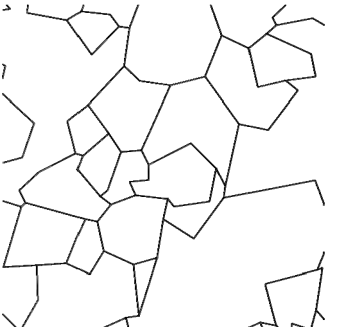
Figure 6 graphically shows some results of a few of the alternatives tested for the land use dataset. Figure 6(a) shows the result of keeping all original coordinates of the boundaries, thus not simplifying them. Tiny details and too many coordinates in the boundaries are the result. It can be seen in Figure 6(b) that the count-stop approach applied on *all* boundaries of the new area leads to a very simplified and coarse version. Both alternatives in which only the merged boundaries are simplified leave more details (see Figure (c) and (d)), where the count-stop approach is a bit more 'aggressive' than the eps-stop approach.

This is also illustrated by the graphs in Figure 7. In each graph, it is shown how many coordinates there are left for the total map, after every generalization step. As expected, the line at the top of the graph is the reference situation, where no coordinates are weeded. As already visually illustrated in Figure 6, it is also clear that the approach, where only the merged boundaries play a role in the simplification, is gentler in removing coordinates compared to when all edges of the new area will be simplified.

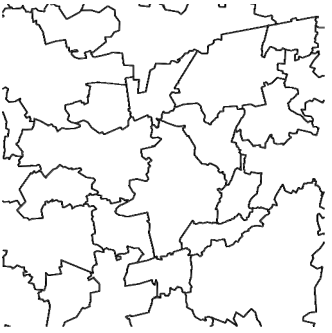
The main cause for this is that if all edges of the new area are simplified, they will be simplified more often, compared to the situation where only the merged edges are simplified (i.e. for every generalization step in which a polygonal area is the area to which a neighbour is merged, its boundary edges will again be simplified).



(a) No simplification (none)



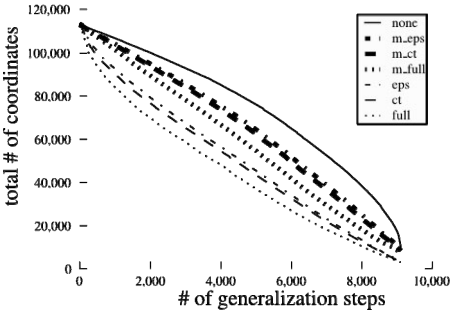
(b) Count stop for all edges of new area (ct)



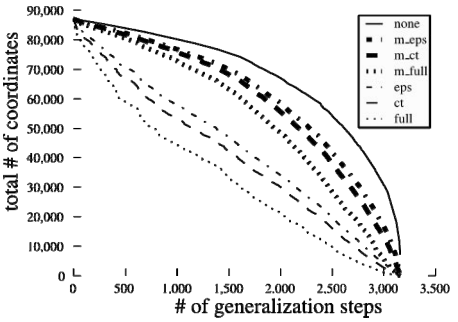
(c) Epsilon stop, using only merged edges (m_eps)

(d) Count stop, using only merged edges (m_ct)

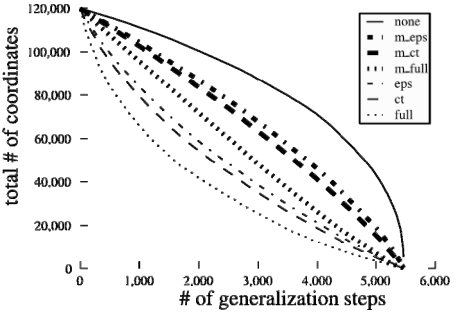
Fig. 6. From the land use dataset: 'Slices' of variable-scale data that show the result of the different alternatives for the line simplification, plotted at the same map scale (within brackets the symbolic name of the tested alternative). Note that the simplification of the boundaries changes the size of the areas and influences the order in which the areas are merged; therefore, the boundaries on the 4 maps do not exactly correspond to each other.



(a) Topographic, urban



(b) Topographic, rural



(c) Land use

Fig. 7. For each dataset, the graph shows the total number of coordinates for the complete map, in each generalization step (i.e. the number of coordinates in a 'slice' of variable-scale data).

Table 3 illustrates the fact that simplifying the boundaries over-and-over again also has a negative effect on the contents of the hierarchy. Although the graphs from Figure 7 show that there are less coordinates on average on every 'slice' derived from the variable-scale structures when all boundaries of a new face are simplified, the opposite is true for the contents of

the data structures. More coordinates need to be stored, because for every line that is simplified, a new version with the simplified geometry also has to be stored in the data structures (e.g. compare alternative 'm_ct' with 'ct'—in all cases more coordinates are stored for the 'ct' alternative). Therefore, simplifying only the merged edges is to be preferred over simplifying all the edges of a new area.

Table 3. Resulting number of polylines in the tGAP hierarchy with their average number of coordinates per polyline and the sum of coordinates in the total hierarchy.

(a) topographic, urban dataset

simplify type	total # polylines	avg # coords per polyline	total # coordinates in hierarchy
None	36,447	7.1	256,969
Ct	60,390	4.3	260,777
Eps	62,006	4.6	284,289
Full	55,084	3.7	205,870
m_ct	36,449	4.6	167,431
m_eps	36,438	4.8	176,350
m_full	36,403	3.8	139,187

(b) topographic, rural dataset

simplify type	total # polylines	avg # coords per polyline	total # coordinates in hierarchy
None	12,347	22.4	276,335
Ct	23,553	8.5	200,860
Eps	24,539	10.1	247,767
Full	19,640	6.7	131,538
M_ct	12,345	11.1	136,940
M_eps	12,343	13.0	160,066
M_full	12,349	7.8	96,665

(c) land use dataset

simplify type	total # polylines	avg # coords per polyline	total # coordinates in hierarchy
None	26,771	15.4	413,250
Ct	54,166	5.8	312,394
Eps	55,603	6.3	348,118
Full	45,040	4.8	216,174
M_ct	26,770	7.5	200,132
M_eps	26,768	8.4	223,623
M_full	26,769	5.3	141,019

6 Conclusion and future work

We described an algorithm to simplify simultaneously a subset of polylines in a planar partition in a variable-scale context. For this, we formalized what we consider a valid variable-scale planar partition. The algorithm is aware not to introduce any topological errors. Furthermore, we gave a theoretical description of the options that we have when employing this algorithm in practice. Another contribution is that we analysed how much the average number of points in the boundaries of the polygonal areas would grow without simplification to choose the best simplification strategy, also from the perspective of the amount of data to be stored in the data structures. Further we showed some visual results.

Some notes on future work:

- We think that an integrated way of formalizing 2D maps plus 1D for scale in a 3D space (leading to 3D volume objects, but where not all axes have the same geometric meaning) could lead to a better axiomatic description of what we consider to be a valid variable-scale hierarchy. This could also lead to an even more continuous variable-scale structure (opposed to our current solution, in which discrete 'jumps' still exist) in which it is possible to gradually morph polylines from the state before applying an aggregation or split operation to the state afterwards (for a technical implementation it might be sufficient to store only the beginning and end states in such a model). As such, it could enable smooth zooming of vector data for an end user.
- We plan to implement the requirements for valid planar partition and vario-scale hierarchy as check constraints in a DBMS (as technical implementation of the conceptual model).

References

Bader M. and Weibel R. (1997) Detecting and resolving size and proximity conflicts in the generalization of polygonal maps. pages 1525–1532.

Barkowsky T., Latecki L. J., and Richter K. F. (2000) Schematizing Maps: Simplification of Geographic Shape by Discrete Curve Evolution. In *Spatial Cognition II*, volume 1849 of *Lecture Notes in Computer Science*, pages 41–53. Springer Berlin / Heidelberg.

Bentley J. L. (1990) K-d trees for semidynamic point sets. In *SCG '90: Proceedings of the sixth annual symposium on Computational geometry*, pages 187–197. ACM, New York, NY, USA.

Da Silva A. C. G. and Wu S. T. (2006) A Robust Strategy for Handling Linear Features in Topologically Consistent Polyline Simplification. In AMV Mon-

teiro and CA Davis, editors, *GeoInfo*, VIII Brazilian Symposium on Geoinformatics, 19-22 November, Campos do Jordão, São Paulo, Brazil, pages 19–34.

De Berg M., Van Kreveld M., and Schirra S. (1998) Topologically Correct Subdivision Simplification Using the Bandwidth Criterion. Cartography and Geographic Information Science, 25:243–257.

Douglas D. H. and Peucker T. K. (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2):112–122.

Dyken C., Dæhlen M., and Sevaldrud T. (2009) Simultaneous curve simplification. Journal of Geographical Systems, 11(3):273–289.

Gröger G. and Plümer L. (1997) Provably correct and complete transaction rules for GIS. In *GIS '97: Proceedings of the 5th ACM international workshop on Advances in geographic information systems*, pages 40–43. ACM, New York, NY, USA.

Guibas L. J. and Sedgewick R. (1978) A dichromatic framework for balanced trees. In *19th Annual Symposium on Foundations of Computer Science, 1978*, pages 8–21.

Kulik L., Duckham M., and Egenhofer M. (2005) Ontology-driven map generalization. Journal of Visual Languages & Computing, 16(3):245–267.

Ledoux H. and Meijers M. (2010) Validation of Planar Partitions Using Constrained Triangulations. In *Proceedings Joint International Conference on Theory, Data Handling and Modelling in GeoSpatial Information Science*, pages 51–55. Hong Kong.

Meijers M., Van Oosterom P., and Quak W. (2009) A Storage and Transfer Efficient Data Structure for Variable Scale Vector Data. In *Advances in GIScience*, Lecture Notes in Geoinformation and Cartography, pages 345–367. Springer Berlin Heidelberg.

Ohori K. A. (2010) Validation and automatic repair of planar partitions using a constrained triangulation. Master's thesis, Delft University of Technology.

Plümer L. and Gröger G. (1997) Achieving integrity in geographic information systems—maps and nested maps. Geoinformatica, 1(4):345–367.

Ramer U. (1972) An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing, 1(3):244–256.

Saalfeld A. (1999) Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm. Cartography and Geographic Information Science, 26:7–18.

Van Oosterom P. (1990) Reactive Data Structures for Geographic Information Systems. Ph.D. thesis, Leiden University.

Van Oosterom P. (2005) Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest. Cartography and Geographic Information Science, 32:331–346.

Visvalingam M. and Whyatt J. D. (1993) Line generalisation by repeated elimination of points. The Cartographic Journal, 30(1):46–51.

Validating a 3D topological structure of a 3D space partition

Bregje Brugman, Theo Tijssen, Peter van Oosterom

Delft University of Technology, OTB, Section GIS-technology,
Delft, The Netherlands.
bregjebrugman@gmail.com, {T.P.M.Tijssen, P.J.M.vanOosterom}
@tudelft.nl

Abstract. The goal of this research is to develop a 3D topological structure to represent a 3D space partition with validation functionality and support for conversions from topological to geometrical primitives. Several 3D topological structures have been presented in the past, mainly by researchers. The technical (implementation) model developed in this paper is based on the conceptual model of the ISO 19107 'spatial schema' standard and consists of four topological primitives: node, edge, face, and volume, which are related to each other via their (co)boundary relationships. In our setting, only linear primitives (no curves) are supported and no isolated and dangling primitives are allowed. In our model, the rings, the shells, and the orientation play key roles within the topological structure and the functions that implement the geometrical realization.

There was no formal definition of a valid 3D topological structure available and this paper presents such a definition, which is the main novel contribution. This definition is presented in three levels, where at every next level the definition is further refined such that finally a set of rules is proposed, which can be implemented unambiguously. In order to validate a 3D topological structure, the involved volumes must be valid as well as the whole structure, which means the relationships between the volumes. The rules for a valid structure have been implemented on top of Oracle Spatial and tested with artificial and real-world test data.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_18,

1 Introduction

The first use of topology has been attributed to Euler in 1736; since then, topology has evolved in mathematics and, more recently, also in GIS. Since the second half of the 20th century, 2D topological data structures are well established with structures like TIGER (Topologically Integrated Geographic Encoding and Referencing system) (Boudriault 1987) and GBF/DIME (Geographic Base File/Dual Independent Map Encoding), both from the US Census Bureau. Several 3D topological structures have been developed as well, most of them by researchers; for example 3D FDS, the 3D Formal Vector Data Structure (Molenaar 1990) and SSS, the Simplified Spatial Schema (Zlatanova 2000). No commercial geo-DBMS has implemented a 3D topological structure yet. 1Spatial is currently developing a 3D topology model as an extension for Oracle Spatial.

A geo-DBMS (Database Management System), where geometric data is stored together with administrative data in one DBMS, is very useful for spatial data management purposes. The main advantages of using an integrated architecture are the capability of a DBMS to handle large volumes of data, the ability to ensure the logical consistency and integrity of data and the ability of multi-user control. Most mainstream DBMSs currently support spatial data types and spatial functions built on these spatial data types. Most spatial data types within a DBMS are defined as single geometries, which describes the geometric primitive in a spatial reference system. For some purposes, like managing data structured in a partition, a topological structure will be more suitable. A topological structure describes the relationships between the primitives (node, edge, face, and/or volume).

Geo-DBMSs are well-developed for 2D spatial data management, but underdeveloped for the third dimension, while 3D spatial data management is becoming more and more important within the 'geo-industry'. For sectors like urban planning, emergency services, hydrology and telecommunication, 3D data management is required. The availability of 3D data is also growing due to new data acquisition techniques. For the past 30 years a lot of research has been carried out and a lot of progress has been made in the field of 3D spatial data management. GEO++ is an early example of a 3D GIS, based on the geo-DBMS Postgres (van Oosterom, Vertegaal and van Hekken 1994). In the commercial sector, Oracle Spatial has implemented a 3D single geometry data type, but most commercial geo-DBMSs have only included 3D coordinates within their single geometries. This means usually that each x,y coordinate has (only) one z-value. This is often referred to as 2,5D. This option of storing 3D coordinates (x,y,z) in the geo-DBMS makes it possible to model 3D with 2D primi-

tives, for example by combining several polygons. Modeling in this way is restricted; some spatial functions do not work, for example the validation of a volumetric object as a whole (Khuan et al. 2008). 3D topological structures are less developed than 3D single geometries.

Before performing any spatial analysis on a topological structure or single geometry, the representation needs to be valid. When data is added or updated the topological rules need to be checked. Therefore a validation function is needed. Current geo-information standards are underdeveloped and not unambiguous with respect to defining a valid 3D topological structure, which leads to different interpretations. Implementation specifications for 3D primitives (both geometrical and topological) are not set yet.

2D validation functions exist for single geometries, but also for topological structures (like Oracle's 2D topological validation function). 3D validation functions also exist but are rather rare and mainly for single geometries, for example Borrmann (2008). No real 3D validation functions for topological structure exist. Only 1Spatial is currently developing a 3D topological structure including a validation function (Watson et al. 2008).

In this article, the implementation of a 3D topological structure with validation functionality will be described. As far as the authors know, this is the first time a validation function is implemented for a 3D topological structure. This article is divided into four sections. First, an overview of relevant work is given in section 2, which is followed by a discussion in section 3 of the formal definition and validation rules which are needed for a valid 3D topological structure. These validation rules have been translated into tests which are implemented as a prototype in Oracle Spatial as described in section 4. Section 5 presents the tests with artificial and real data. Finally, the conclusion in section 6 contains the discussion about the main contributions of this work and suggestions for future research.

2 Review of related work

Validation functions are characterized by the moment of validation, the user influence, and the validation rules. The validation rules are the most important aspect of validation and are based on the definition of a valid primitive/structure. In the case of the 3D primitives, no geo-information standards are yet available. 2D single geometries are standardized, including implementation specifications, in the Simple Feature Specification of the OGC and ISO (OGC 05-126/134:2005 and ISO 19125:2004). The OGC has published a candidate version in 2006 with a corrigendum, correcting editorial and minor technical issues in 2010. In this version, still

only 2D geometrical primitives are discussed, but possibly within a 3D context (x,y,z coordinates). The Complex Feature Specification, the OGC standard (at implementation specification level), for topological primitives and structure is not finished yet. For 3D primitives (topological and geometrical), there is an abstract ISO specification (ISO 19107:2003 Geographic information — Spatial schema), but this is not the needed implementation specification. Even the available 2D geo-information standards, however, are ambiguous and incomplete as Van Oosterom et al. (2004) pointed out for simple polygons. Also the interpretations of different GIS and DBMS vendors differ from each other and from the standards. As an introduction to the proposed 3D topological structure, Oracle's 3D single geometry (subsection 2.1) and 1Spatial's 3D topology (subsection 2.2) will be described briefly below. In the context of 3D city models, Gröger and Plümer (2009) present a set of axioms for valid models. A drawback is that they do not allow non-2-manifold objects; see Figure 1 top-left object. Further, they do not consider a space partition, but a city model, that is, a collection of 3D volume objects related to the Earth surface (with a lot of unmodeled empty space).

2.1 3D single geometry (Oracle)

The geometrical equivalents of the topological primitives are the point, line, polygon and solid. Oracle has implemented a geometrical 3D primitive called the 'simple solid' (single geometry) since the release of Oracle Database 11g. The definition of this simple solid: 'a 'single volume' bounded on the exterior by one outer bounding surface and on the interior by zero or more inner bounding surfaces. To demarcate the interior of the solid from the exterior, the 3D-polygons of the outer bounding surface are oriented such that their normal vector always point 'outward' from the solid. In addition, each 3D-polygon of the bounding surfaces has only an outer boundary and no inner boundary (Kazar et al. 2008). Oracle also included a validation function for these simple solids with a set of predefined rules. The function checks for type consistency and geometry consistency (Murray et al. 2010).

Determining the validity of a solid is not always easy. A solid can be connected and closed but still not bound a single volume. This is related to the number of times an edge is used in a solid. An edge in a solid can be used more than two times (as long as it is an even number) and still remain valid; see Figure 1. Furthermore it is not enough to test the inner and outer bounding surfaces only for intersections. An outer bounding surface could

be completely covered by an inner bounding surface without touching each other.

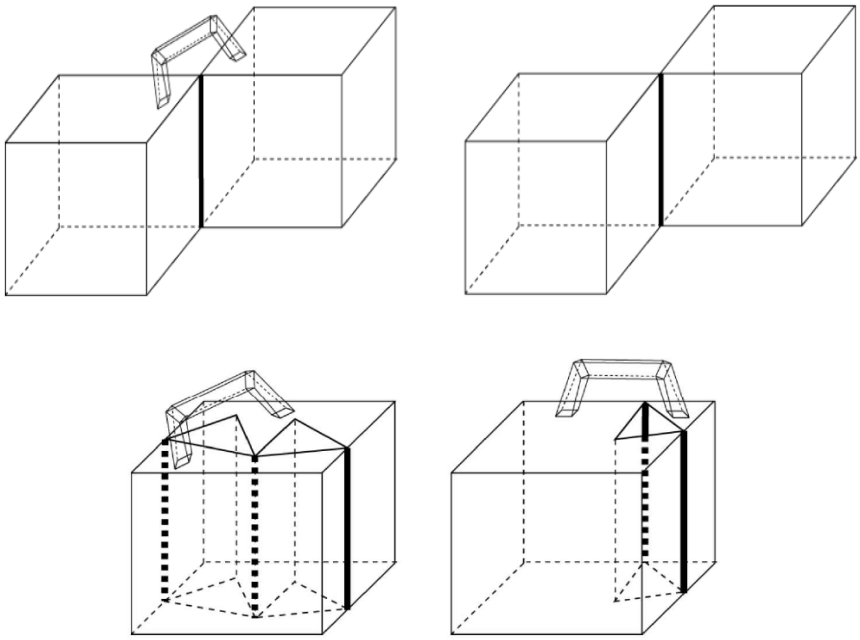


Fig. 1. Top left a valid simple solid; top right an invalid simple solid; bottom left a valid simple solid; and bottom right a valid simple solid; Note the thick edge is used 4 times (source: Kazar et al. 2008)

2.2 3D topology (1Spatial)

At this moment, Radius Topology is only available in 2D, but 1Spatial is currently performing research driven development for a 3D variant. An implementation method describing how to implement a user defined geometry into the topological structure of Radius Topology is available (Watson et al. 2008). The implementation of the topological structure in Oracle Spatial consists of four primitive tables (one for each primitive) including explicit storage of the geometry of the nodes, edges, and faces and three linking tables storing the relationships between the primitives including their orientation.

The 1Spatial validation rules are not enough to test for a valid 3D topological partition. First, the validation rules apply to single primitives and not to the whole structure; intersecting volumes will not be detected. Sec-

ond, these rules are not enough to guarantee valid 3D primitives (e.g. solids are not checked for a contiguous volume or proper orientation).

3 Validation rules for 3D topology structures

The main difference between validating single geometries and validating topological structures is the relationship between the 3D primitives. Topological primitives (volumes) are part of a whole structure, while the single geometries 'stand alone'; they are validated as one object, only dealing with their internal geometrical characteristics. A topological primitive must be valid on its own and valid within the structure. The explicit relationship between neighbouring primitives is one of the advantages and characteristics of a topological structure. A topological structure is less appropriate when there are no (direct) neighbours around.

The topological structure will be validated according to a predefined set of rules. Before defining a valid structure, some initial conditions will be set. The designed 3D topological structure represents a full space partition of volumes. The volumes are represented by their boundaries, the structure will be linear, and finally the structure will be ISO 19107 compliant; see Figure 2. The top level definition of a valid structure: '*a topological space which is divided into a set of non-overlapping valid linear volumes without any gaps*'. In the second level of detail of this definition, four aspects can be distinguished:

a. The *topological space* is defined by 1 or more inner shells of the universal volume, which has no outer shell containing the inner shell(s). These inner shells must be valid shells and are not allowed to intersect (touch is allowed).
b. A *set of non-overlapping volumes without any gaps* means that every face is on the boundary of exactly two volumes, one on each side. Furthermore volumes are not allowed to intersect and no isolated and dangling primitives are allowed.
c. *Linear volumes* can be established by planar faces and straight edges.
d. The definition of a *valid volume* is based on existing definitions of valid solids. Each volume must consist of one outer shell and zero or more inner shells. Each shell consists of 4 or more valid faces, which are non-overlapping, properly oriented, and connected in a topological cycle. Shells are allowed to touch each other or themselves by node or edge (not by face). The geometric realization of each volume bounds a single volume.

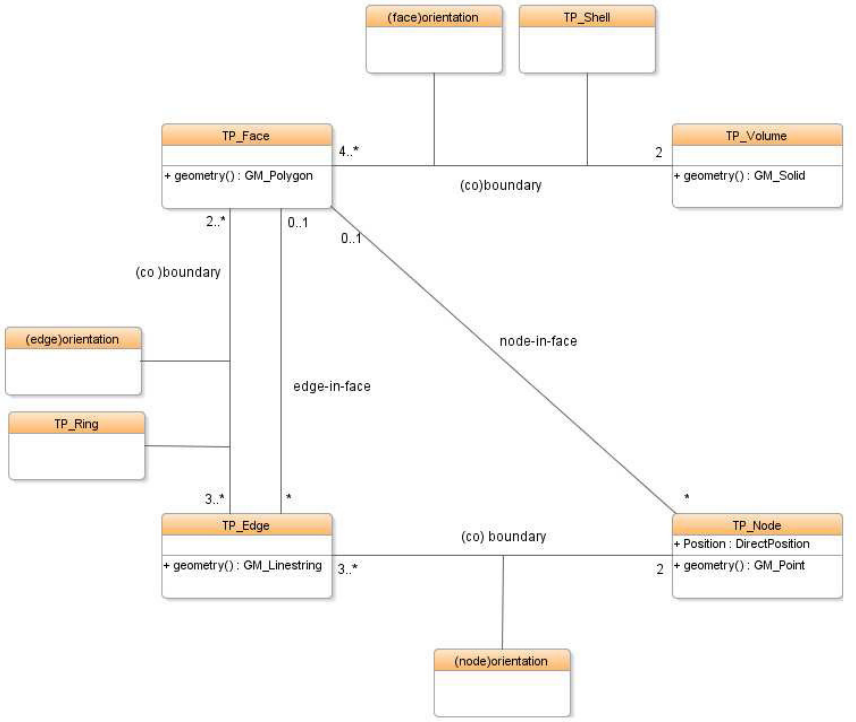


Fig. 2. The conceptual schema of our 3D topological structure (based on ISO 19107)

To meet the above definition, the structure needs to meet the following rules, which form the third and most detailed level of our definition of a valid 3D topological structure. Some rules deal with topology, others are more geometrical in nature. All rules are explained below.

1. *Unique primitives:* Each node has unique xyz values (within the geometrical tolerance distance), which hold the coordinates of a node. When each node is unique, each (unique) combination of primitives forming another primitive is unique as well.
2. *Each primitive is a volume or part of the boundary of a volume:* Because the structure consists of only volumes without isolated primitives, each primitive must be part of a volume. Shells and rings can be added to this rule: each node is part of an edge, each edge is part of a ring, each ring is part of a face, each face is part of a shell, and each shell is part of a volume.
3. *Each undirected primitive is associated with two opposite directed primitives:* Since the structure is a full space partition and no isolated

and dangling primitives are allowed (rules 2 and 5), every primitive is associated with two opposite directed primitives of the same dimension. Each volume has a neighbour on each side; this means each face is associated with two directed faces. Since each primitive is part of a volume (rule 2), it is sufficient to only test the faces.

4. *Proper orientation:* Rule 3 requires that every primitive has a positive and a negative direction, but the topological structure does not only require an orientation but also requires a proper orientation. This means that every face part of a shell is oriented outward from the volume it bounds. In this way the interior of a volume can be distinguished from the exterior. The orientation of an inner ring of a face must be opposite from the orientation of the outer ring.
5. *Each boundary is closed:* When each boundary is closed and each primitive is part of a volume (rule 2), it means no dangling primitives (edges and faces) are present in the structure. The boundary of a volume is specified by its shells, the boundary of a face by its rings, the boundary of an edge by its nodes. Each volume must consist of one outer shell and zero or more inner shells, where the inner shells are directly inside the outer shell. Each shell must be closed, which is the case when each edge in the shell is on the boundary of two or more (even number) faces (of the same shell). Each face must contain one outer ring and zero or more inner rings, where the inner rings are directly inside the outer ring. Each ring consists of 3 or more edges and must be closed. A closed ring is established when all edges are connected in a topological cycle. This means each node within the ring is part of two directed edges (within the ring), once as start node and once as end node. Each edge has a start node and an end node.
6. *Valid extent:* The 3D objects that together make up a data set are located in the universal volume, a volume without an outer shell. The extent of the data set is comprised of one or more inner shells of the universal volume. A ‘hole’ in the data set could be modeled as a small outer shell of the universal volume (contained within a larger inner shell of the universal volume).
7. *The structure is linear:* A linear structure (no curves) is one of the preset conditions; therefore it needs to be checked. When all faces are planar and all edges are straight, the structure is linear. When a face is planar, it means all nodes of that face are on the same plane (within a geometrical tolerance value). This topological structure only uses straight edges, defined by a start and end node (so the edges do not include intermediate vertices).

8. *Inner boundaries must be inside outer boundaries:* Every inner shell must be inside the outer shell of the same volume. Inner shells are not allowed to be inside other inner shells of the same volume. The same applies for inner rings. Inner rings must be inside the outer ring of the same face and are not allowed to be inside other inner rings of the same face.
9. *No intersections:* No intersections between and within shells and rings are allowed. No intersections between and within shells means no intersection of faces (within one shell or between different shells), but faces are allowed to touch (when a node or edge is present at the intersection or when it is about a node-in-face or edge-in-face singularity; see Figure 3). No intersection between rings is allowed, although they are allowed to touch when a node or edge is present. An inner ring is allowed to touch the outer ring in one point (of the same face). Outer and inner rings are not allowed to touch themselves.
10. *Bounding single volumes/areas:* Every volume should have a contiguous interior and every face should have a contiguous interior. The boundaries are already checked for being closed (rule 5). All primitives involved in the boundary are connected to each other (rule 2), but this does not guarantee that the interior is contiguous; see Figure 1 top-right. A more specific test is needed for this purpose.

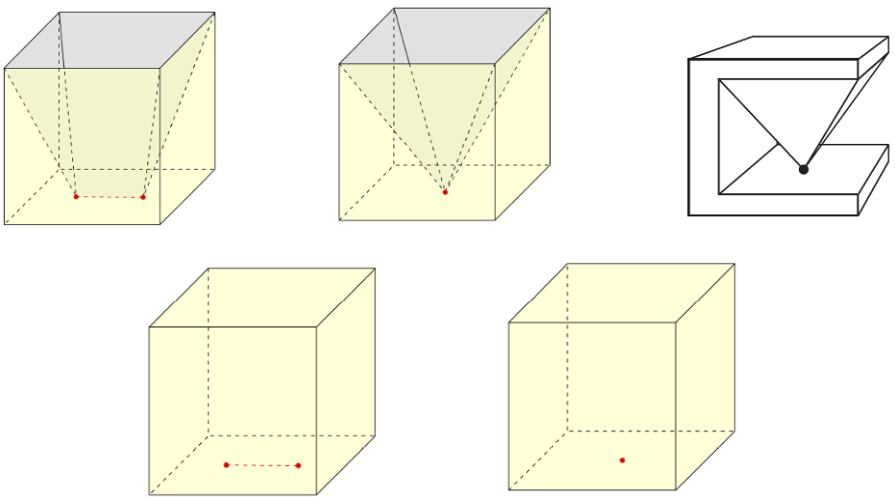


Fig. 3. Top row: allowed singularities; top-right example from Borrmann (2008; see page 218, fig 9.6 right); bottom row: invalid singularities

Some additional notes can be made related to rule 9 and the singularities. That is, lower dimensional primitives completely inside the interior of a higher dimensional one; e.g. node-in-edge, node-in-face, node-in-volume, edge-in-face, edge-in-volume, and face-in-volume. These singularities are only allowed if they are needed to define a volume in the context of modeling space partitions. Therefore, the face-in-volume, edge-in-volume, and node-in-volume are per definition invalid. Further, the node-in-edge singularity should be treated (remodeled) as a split of the edge as this has no drawbacks. Therefore, it is not allowed to have the node-in-edge singularity. In the UML model (Figure 2), these singularities are not included as associations and only the remaining allowed singularities, edge-in-face and node-in-face, are included.

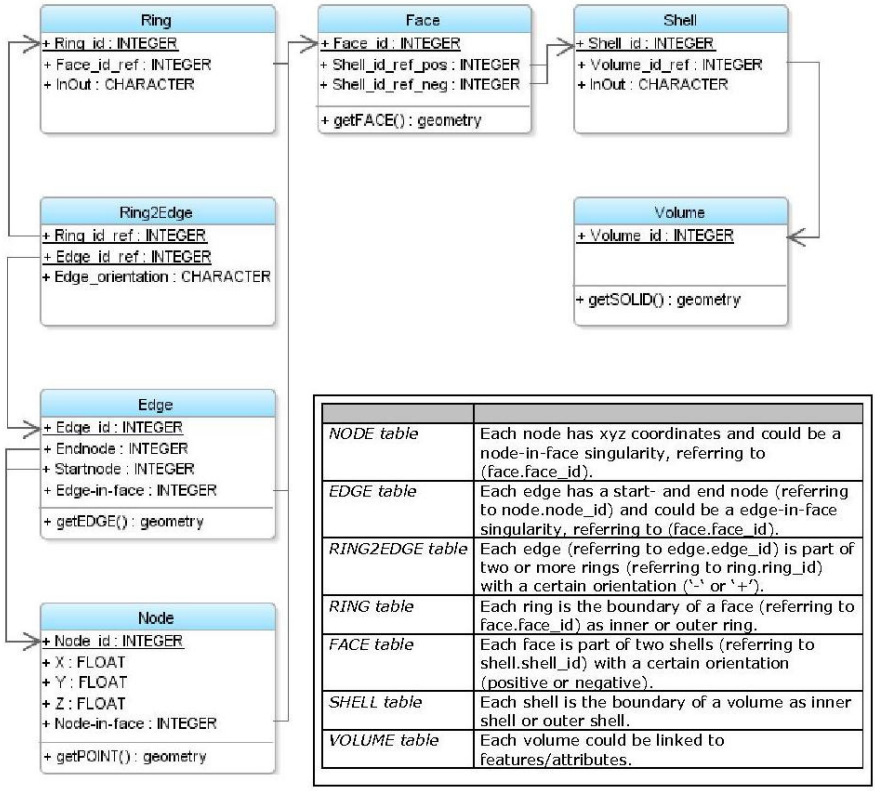
4 Implementation

Different approaches exist in designing a structure. A balance has to be found between little redundancy,(relatively) easy validation tests and geometry operations. If a lot of redundant information is stored, the structure is large and slow and more chances for storing contradictions are present. On the other hand, it will be easier to extract information from the structure (less derivations needed). If, on the other hand, little redundant information is stored, it will require more complex algorithms to retrieve information and it will be difficult to check if the structure is still valid after an edit. This topological structure will consist of manageable tables with little redundancy; a balance is aimed for between simplicity and usability; see Figure 4.

4.1 Conversion functions

For various reasons, it is useful to have conversion functions that ‘materialize’ the topological primitives into geometrical objects. An obvious one is visualization; virtually all display engines require geometry as input, and they are not designed to make ‘direct’ use of topological structures. Although topology certainly has advantages for storage and maintenance of large data sets, many applications also need geometry as input for their operations. Certain GIS operations benefit from being executed in the topology domain (e.g. finding the neighborhood of an object); others perform better or simply can only be executed on geometry objects (e.g. length, area and volume calculations). Conversion functions in the other direction, from ‘raw’ (geometry) data to topological structure, would be very helpful

to fill the structure, but that was not part of the current research. The efforts of 1Spatial as described before can be considered complementary to this research; their system concentrates on generating clean topology from raw data.



NODE table	Each node has xyz coordinates and could be a node-in-face singularity, referring to (face.face_id).
EDGE table	Each edge has a start- and end node (referring to node.node_id) and could be a edge-in-face singularity, referring to (face.face_id).
RING2EDGE table	Each edge (referring to edge.edge_id) is part of two or more rings (referring to ring.ring_id) with a certain orientation ('-' or '+').
RING table	Each ring is the boundary of a face (referring to face.face_id) as inner or outer ring.
FACE table	Each face is part of two shells (referring to shell.shell_id) with a certain orientation (positive or negative).
SHELL table	Each shell is the boundary of a volume as inner shell or outer shell.
VOLUME table	Each volume could be linked to features/attributes.

Fig. 4. The logical (near technical) schema of our 3D topological structure

The to-geometry conversion functions developed coincide with the 3D primitives: getPOINT(id), getLINE(id), getPOLYGON(id), and getSOLID(id). The difficulty of implementation of these functions ranges from trivial, the first three, to rather complex, the last one. The structure, as presented in this paper, makes clear that the geometry of points, lines, and polygons can simply be 'looked up' in the relevant tables. The getSOLID function is more difficult because Oracle geometry is the target for the geometry object. The 3D polygons that make up an Oracle solid cannot contain inner rings, but in our model inner rings are allowed. So each face that contains inner rings has to be split up into a set of planar polygons

without inner rings. As an example below a PL/SQL snippet from the getPOLYGON function:

```
--select outer ring and inner rings
SELECT ring_id INTO v_Oring FROM ring
  WHERE face_id_ref=i_face AND InOut='O';
SELECT ringordinates INTO v_ordinate FROM
  ringordinates WHERE ring_id=v_Oring;
SELECT ring_id BULK COLLECT INTO tab_Iring FROM
  ring WHERE face_id_ref=i_face AND InOut='I';
v_info_array:=sdo_elem_info_array(1,1003,1);
-- if no inner rings present
IF tab_Iring.count=0 THEN
  FACE:=SDO_GEOMETRY(3003,NULL,NULL,
    v_info_array, v_ordinate);
  RETURN FACE;
-- if inner rings are present:
ELSE
  FOR i IN tab_Iring.FIRST..tab_Iring.LAST LOOP
    v_Iring:=tab_Iring(i);
    SELECT ringordinates INTO v_Iordinate FROM
      ringordinates WHERE ring_id=v_Iring;
    ...
  END LOOP;
```

4.2 Validation checks

Orientation and boundaries (especially the groupings in rings and shells) play an important role in topology and validation. Therefore orientation and boundaries are stored explicitly, although it means extra storage and extra validation tests, but this is in no proportion to the advantages. The orientation of edges is stored explicitly by references to a start and end node. The implementation is done in Oracle Spatial, therefore Oracle (proprietary) data types are used. A topological structure has its strengths (and weaknesses) compared to a geometrical model. This means that validation tests based on geometrical characteristics, which is inevitable, will be reduced to a minimum and the validation tests will be based, as much as possible, on the topological relationships between the nodes, edges, faces, and volumes.

Simple (fast) tests will be done on the whole structure and reduce the hard (time-consuming) tests as much as possible. Some constraints are enforced by the use of primary and foreign keys and ‘non-nullable’ columns. Tolerance values play an important role in validation. In all tests, which

require a tolerance value, tolerance values are applied. Table 1 gives an overview of the relationship between the ten validation rules and the eight validation tests.

Table 1. Validation rules linked to the validation tests

	Rule	Test
1.	Unique primitives	1
2.	Each primitive is a volume or part of the boundary of a volume	2
3.	Each undirected primitive is associated with two opposite directed primitives	n.a.
4.	Proper orientation	5/4
5.	Each boundary is closed	¾
6.	Valid extent	3
7.	The structure is linear	6
8.	Inner boundaries must be inside outer boundaries	7
9.	No intersections	7
10.	Bounding single volumes/areas	2/8

Rule 3 is automatically implemented by the columns in the face table (and the primary key on the face_id). Therefore, no test is needed for testing 'each face is associated with two (opposite) directed faces.' The other tests are organized in blocks: first, block I (uniqueness and references) is performed; if successful then block II (structure and orientation) is performed; and finally block III (geometry) is performed, only if I and II are both successful.

Block I, test 1) unique primitives: In this test, nodes, edges, and faces are tested for their uniqueness. Unique volumes do not need to be tested because volumes cannot refer to the same shell (by the primary-key on the shell) and all shells are unique (if all faces are unique). Although all primitives have unique id's (enforced by primary keys), they still need to be tested for their uniqueness. To test if nodes are unique they are stored as point geometries in a temporary table and a spatial index is created. Then each point is checked whether other points can be found inside the tolerance distance from the point using the spatial index to speed up the procedure.

Block I, test 2) primitive references: In this test, the references between the primitives (one dimension higher/lower) will be tested; node-edge, edge-face, and face-volume, in order to avoid isolated (no reference to a

primitive in a higher dimension) and nonexistent (no reference to a primitive in a lower dimension) primitives. This check does not exclude dangling edges and faces, which will be checked in test 4. Checking the references is partly taken care of by not allowing NULL values in the tables and applying foreign keys, but it is necessary to perform some additional tests for isolated nodes and edges and for nonexistent faces, rings, and shells. The primitives are also tested for their minimum number of boundary primitives: a volume must consist of 4 or more faces and a face must consist of 3 or more edges. An edge must consist of exactly two nodes, but this is taken care of by the columns start node and end node in the EDGE table and therefore does not need to be tested. The third aspect in this test deals with primitives used more than two times in one boundary. A node could be on the boundary of many edges, but an edge can only be used once in the boundary of a face (irrespectively of being an inner or outer ring) and a face can only be used once on the boundary of volume.

Block II, test 3) each face/volume consists of one outer boundary: Each face is tested for exactly 1 outer ring (no more and no less than 1 outer ring) and each volume for 1 outer shell. The universal volume is an exception.

Block II, test 4) closed boundaries: Each boundary has to be closed. The boundaries of faces and volumes are checked by testing for closed rings and shells (by the orientation of the edges and faces within the rings and shells). A closed ring is established when the last node equals the first node. A closed shell is established when each edge in the shell is on the boundary of two or more (even number of) faces (of the same shell). This means that each edge, involved in the shell, must have an equal distribution of negative and positive references.

Block II, test 5) proper orientation: Every face in a shell is oriented outward (positive). For the orientation of rings within a face, the outer ring should be counter clockwise when observed from outside the volume (normal of vector will point outward) and the orientation of an inner ring must be opposite from the orientation of the outer ring (of the same face).

Block III, test 6) planar faces: A face is planar when all nodes of that face are on the same plane (within a planarity tolerance value). In this test, an optimal plane is fitted through the points of the face and then the distance of each point to the optimal plane is calculated. The test fails if the distance is bigger than the tolerance value. This is currently implemented by taking the plane defined by two arbitrary edges of the outer ring and checking if the 'average' of nodes is on the plane (first the outer ring is tested, next the inner rings). This is not a very good test as with a bit of luck one could pass this test as a point far above the plane is averaged out by a point far below the plane. A better test would be to first find the ‘best

fitting' plane of a face by the least-squares error fitting method on all boundary nodes (of outer and inner rings) and nodes related to proper singularities in this face (node-in-face and via edge-in-face). Then check if all nodes are within the tolerance distance of the plane. One remaining issue is: how is the tolerance defined? Some possible options are: a. fixed value (but this would mean that a very large face has to be relatively flat compared to a small face); b. relative value, that is, expressed as the amount of curvature allowed (fraction of the diameter of the face; with diameter face defined as the distance between the two points most far apart).

Block III, test 7) no intersections: Primitives are not allowed to intersect but are allowed to touch. This can be summarized in 'volumes are not allowed to intersect and self-intersect (but touch)', which can be translated into 'faces are not allowed to intersect and self-intersect (but touch)'. When all volumes are not self-intersecting, no intersecting volumes will be present because the structure is a full space partition (and each face is part of exactly two different shells). Therefore, volumes do not need to be tested on intersections between each other but each volume has to be tested on self-intersection (including the inner shells of the universal volume). This will be done by testing each volume for intersecting faces. In four particular cases, touching faces are valid; at an edge-sharing or node-sharing touch and at an edge-in-face or a node-in-face singularity. Self-intersecting faces will always lead to a face intersection somewhere in the volume, therefore no separate test for self-intersecting faces is needed. Inner rings are allowed to touch their outer ring (in one node only). In addition, inner rings and inner shells need to be tested for not being completely outside the accompanying outer ring or outer shell (except for the universal volume) and outer rings and outer shells need to be tested for not being completely covered by their accompanying inner ring or inner shell (except for the universal volume with a 'hole', which is modeled as a small outer shell). This test makes use of Oracle's SDO_ANYINTERACT (which is fitted with a tolerance value).

Block III, test 8) bounding single volumes/areas: Each volume and each face must have a single, contiguous interior. Most volumes will have a single interior if they have passed tests 1 through 7. However, some complicated situations make this last test difficult, see Figure 1 for examples. A solution for this problem is suggested by Kazar et al. (2008). They suggest the tetrahedronization of a volume. Next, all connected tetrahedra will be marked via shared triangles (by a Boolean value) starting at a random tetrahedron. At the end, the number of marked tetrahedra must be equal to the total number of tetrahedra, otherwise the volume is not contiguous. This test has not been implemented in the current research.

All tests are written as PL/SQL scripts. At the moment the validation tests run on the whole dataset (global testing), but internally they test each object separately therefore it could be very useful for a local test as well (for example after an edit/update).

5 Test with real data

For testing the prototype, a topological correct and clean data set of the TU Delft Campus has been used: 370 buildings, from the 2D Large Scale Base map of Delft (Grootschalige Basiskaart in Dutch), which were extruded to 3D; see Figure 5 (Ledoux and Meijers 2011). The original data was converted to the 3D topological structure (Brugman 2010). First, the data has been analyzed and inserted into the prototype, then the validation tests and geometry operations have been tested on the data set. The data consist of 370 (unique) volumes and 8152 (unique) faces. Furthermore 13467 (unique) edges and 5841 (unique) nodes could be derived. When analyzing the data in more detail, the following information could be extracted: the volumes (buildings) are scattered around the area (campus) and the air or the underground are not explicitly modeled in this case. When clustering the volumes, 169 clusters can be distinguished. A cluster consists of buildings that are connected to each other; connected buildings share at least one face. Different clusters are separated from each other by 'air'.

The data was inserted into the tables of the prototype. In order to present a full space partition, a 'mini' universe has been created for each cluster. This means the universal volume has 169 inner shells. The validation tests and geometry operations are tested on the whole campus data set and on the largest cluster (cluster 381) with 33 buildings; see Figure 6. This cluster consists of 33 volumes and the universal volume with one inner shell, 893 faces, 1455 edges, and 594 nodes. At the moment, test 7 (no intersections) is the most time consuming, but with the help of spatial index structures this can be improved (but performance was not in the scope of the current research).

Bentley's MicroStation, the software used for the visualization, does not know of or 'understand' the 3D topological structure. Visualization was achieved through the to-geometry conversion functions, getPOLYGON in particular, that produces geometries that can be visualized with a CAD/GIS-like MicroStation.

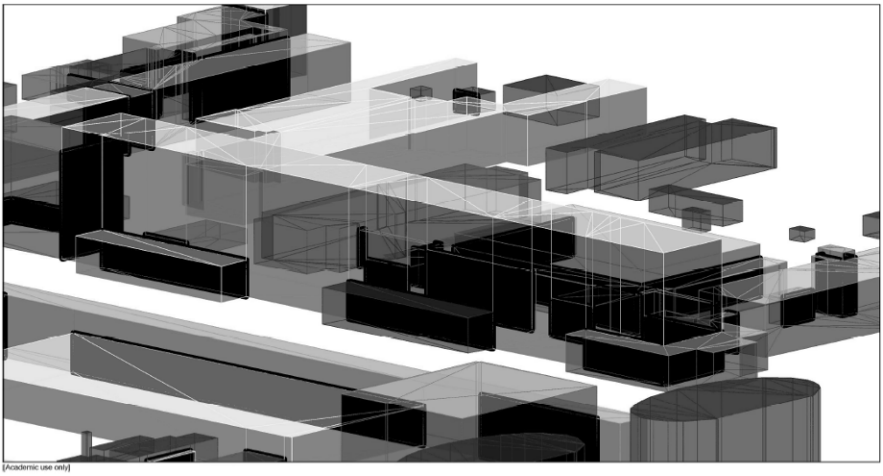


Fig. 5. Several buildings with shared faces (shown in black; visualized with MicroStation)

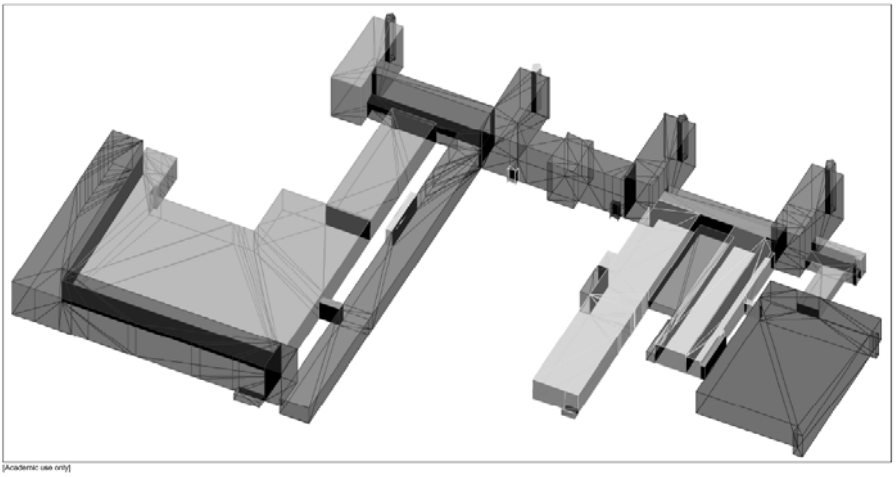


Fig. 6. Cluster 381 of the Campus data set (visualized with MicroStation)

Some tests could not be carried out because the data set contained no inner shells and no inner rings. These were tested on a hand-made, artificial mini data set, which consists of only 10 volumes (plus one universal volume), 58 faces, 110 edges, and 67 nodes; see Figure 7.

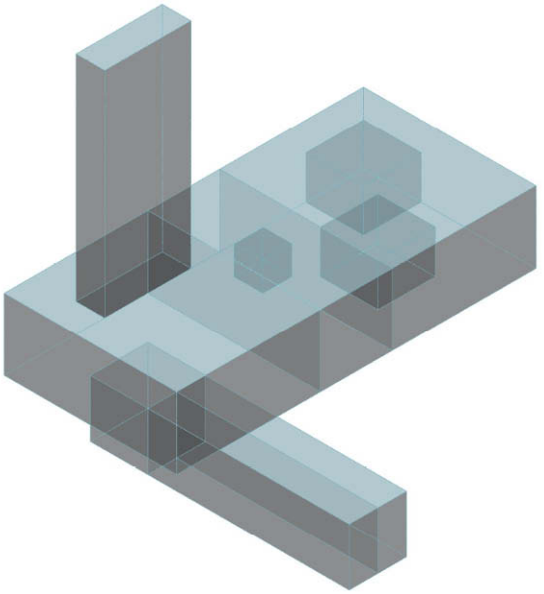


Fig. 7. Artificial mini data set (visualized with MicroStation)

6 Conclusion

The most important novel contributions of this paper are all included in the developed prototype (developed in Oracle Spatial using PL/SQL):

- Table structure for a balanced implementation of a 3D topological structure based on ISO 19107;
- Validation tests, based on a 3-level formal definition of a valid structure; and
- Conversion functions to sdo_geometry, i.e. the geometrical realization with four TO_GEOMETRY operations: 'Node to Point', 'Edge to Line', 'Face to Polygon', and 'Volume to Solid'.

The current implementation is limited to linear primitives (no curves) and the following topics are outside the scope of the current research (but within the scope of future research): updating/editing, modeling features, and efficient validation tests/queries. Also, in the future test 8 'contiguous volume' should be implemented. The important open question is: are the 10 validation rules sufficient and are they all needed?

Other future work could include automatic repair of invalid situations, e.g. non-flat faces. Assume a non-flat face is detected during validation,

but that it is further well-connected in the structure. This can be repaired by splitting it into two (or more) faces. Normally, splitting a face should be a local operation and keeps the topology structure intact as it does not create new intersections. After the split check both faces again for flatness and repeat splitting until the faces are flat enough. For sure this process will terminate, because if you continue until you get triangles, then these will always fulfill the criterion of flatness. However, when trying to minimize the number of resulting faces, the procedure will be non-trivial. What is an optimal split, i.e. removing most of the 'tension' in the non-flat face? Also the proper handling of holes (inner rings) must be taken care of. Instead of this top-down approach, perhaps a bottom-up alternative approach would be easier: 1. triangulate face (including inner rings), 2. merge neighbor triangles if ‘merger’ is flat enough, and 3. stop when no more ‘mergers’ are possible. However, it is again not clear in which way a minimum number of flat faces can be obtained.

Acknowledgments

The authors of this paper would like to express their sincere gratitude to our colleagues Rod Thompson, Hugo Ledoux, and to the three anonymous AGILE reviewers for their constructive remarks and suggestions.

References

Borrmann A. (2008) Computerunterstützung verteilt-kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken. PhD thesis Technische Universität München.

Brugman, B. (2010) 3D topological structure management within a DBMS - validating a topological volume. GIMA Master's Thesis

Gröger, G. and Plümer, L. (2009) How to achieve consistency for 3D city model. GeoInformatica, DOI: 10.1007/s10707-009-0091-6.

OGC (2005), OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 1: Common architecture (OGC 05-126) / Part 2: SQL option (OGC 05-134), version 1.1.0.

ISO/TC211 (2003) International standard ISO 19107, Geographic information – spatial schema. ISO: Geneva, Switzerland

Kazar, B.M., R. Kothuri, P. van Oosterom & S. Ravada (2008) On valid and invalid three dimensional geometries. Advances in 3D Geoinformation Systems, Springer, pp. 19-46

Khuan, C.T., A. Abdul Rahman & S. Zlatanova (2008) 3D solids and their management in DBMS. Advances in 3D Geoinformation Systems, Springer, 2008, pp. 279-311

Ledoux, H. and Meijers, M. (2011) Topologically consistent 3D city models obtained by extrusion. International Journal of Geographical Information Science. In Press.

Molenaar, M. (1990) A formal data structure for 3D vector maps. Proceedings of EGIS'90, Vol. 2. Amsterdam, The Netherlands, pp. 770–781

Murray, C., D. Abugov, N. Alexander, B. Blackwell, R. Chatterjee, D. Geringer, M. Horhammer, Y. Hu, B. Kazar, R. Kothuri, S. Ravada, J. Wang, J. Yang. (2010) Oracle Spatial Developer's Guide 11g Release 2 (11.2)

Oosterom, P. van, W. Quak & T. Tijssen (2004) About invalid, valid and clean polygons. Peter F. Fisher(Ed.); Developments in Spatial Data Handling, 11th International Symposium on Spatial Data Handling, 2004, pp. 1-16

Oosterom, P. van, W. Vertegaal & M. van Hekken (1994) Integrated 3D modelling within a GIS. AGDM'94: Delft, Netherlands

Watson, P.J., M.J. Martin & T.D.c. Bevan (2008) WO 2008/138002 A1 Three-dimensional topology building method and system. World Intellectual Property Organization: Geneva, Switzerland

Zlatanova, S. (2000) 3D GIS for urban development. PhD thesis, ITC, The Netherlands, 222pp

Querying Vague Spatial Information in Geographic Data Warehouses

Thiago Luís Lopes Siqueira[1,2], Rodrigo Costa Mateus[3], Ricardo Rodrigues Ciferri[2], Valéria Cesário Times[3], Cristina Dutra de Aguiar Ciferri[4]

[1]São Paulo Federal Institute of Education, Science and Technology, IFSP, São Carlos, Brazil
[2]Computer Science Department, Federal University of São Carlos, UFSCar, São Carlos, Brazil
[3]Informatics Center, Federal University of Pernambuco, UFPE, Recife, Brazil
[4]Computer Science Department, University of São Paulo at São Carlos, USP, São Carlos, Brazil
prof.thiago@cefetsp.br, rcm3@cin.ufpe.br, ricardo@dc.ufscar.br, vct@cin.ufpe.br, cdac@icmc.usp.br

Abstract. Non-redundant geographic data warehouse (GDW) schemas have been recognized as an essential issue in the GDW design. However, little attention has been devoted to the study of how the handling of vague spatial data affects query performance and storage requirements in GDW. In this paper we investigate the query processing performance over non-redundant GDW schemas that are based on different spatial representation approaches for handling spatial data uncertainty. Further, we analyze the indexing issue, aiming at improving query performance on a non-redundant GDW with vague spatial data. We concluded that the adaptation of an existing index for GDW aiming at handling uncertain spatial data does not satisfy completely the performance requirements. Therefore, there is a need for new index structures for processing vague objects in GDW.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_19,

1 Introduction

The provision of decision-making support has drawn much interest from researchers in Geographic Information System (GIS), Data Warehouse (DW) and On-Line Analytical Processing (OLAP). Several studies have been developed for providing decision-making users with tools capable of carrying out spatial analysis together with multidimensional analytical queries over huge historical data sets. Usually, this is achieved by making use of a Geographic Data Warehouse (GDW) (Ferreira et al., 2001; David, Somodevilla, and Pineda, 2007; Malinowski and Zimányi, 2008; Times et al., 2008). However, most of the research has focused on the representation of crisp spatial objects in GDW, i.e. objects represented by a well-defined geometry. Little attention has been devoted to both the experimental evaluation of query processing performance and the measurement of storage requirements of a GDW with the ability to handle geographic phenomena with uncertain location or regions with undetermined boundaries, i.e. GDW with vague spatial objects.

Handling vague spatial objects in GDW implies a need for designing spatial dimensions that store vector geometries denoting uncertain locations or regions with undetermined boundaries, as well as for processing spatial and multidimensional queries to allow users to request information. In order to investigate the effects of storing vague spatial representations in GDW, in this work, a spatial dimension with vague spatial information contains a vague spatial attribute that extends a conventional spatial attribute by storing a set of vector geometries instead of storing a single geometry. For instance, a crisp region denoting the location of a given farm and represented as a single polygon may be extended to two concentric polygons to model a spatial region with undetermined boundaries. These two polygons provide limits to define the range of indeterminacy and follow the Egg-Yolk representation model (Cohn and Gotts 1996). Another example is given when uncertainty related to a client address is modeled as a set of points to represent all possible client locations.

Clearly, the representation of vague spatial locations may lead to an increase in the storage costs of vector geometries. Regarding query processing, the challenge in GDW with vague spatial objects is to compute spatial predicates among ad hoc spatial query windows (that can be a crisp rectangle or a rectangle with undetermined boundaries) and spatial objects with vague spatial representations. Therefore, the query processing in such GDW may not lead to the same results obtained in a strictly crisp GDW, due to their different spatial representations. Then, an experimental evaluation approach should aid GDW designers to improve the data schema qual-

ity and, consequently the performance of their systems. In this paper we add uncertainty to GDW by storing a set of vector geometries that denote spatial vagueness (called here **vague GDW**), analyze the DBMS (database management system) performance according to different types of vague spatial representations, and assess the feasibility of employing an existing index for vague GDW in order to enhance the query processing performance.

The literature states that spatial dimension tables must not contain redundant data (i.e. repeated geometries) in order to avoid high performance losses both in query processing and in storage requirements of GDW (Siqueira et al., 2009). Moreover, the chosen representation for spatial objects and the inclusion of the spatial attribute together with or separate from the conventional dimension table affect query processing in GDW (Mateus et al., 2010). Therefore, an experimental evaluation of spatial queries over non-redundant GDW schemas that are based on different vague spatial representation approaches for handling uncertainty is needed for helping the design of logical GDW schemas. The motivation behind performing such evaluation is given as follows. Although avoiding spatial data redundancy decreases storage requirements, it implies a need for performing expensive additional join operations to answer a given query that may refer to one or more spatial query windows. In this paper, we also examine if the complexity of the spatial objects for representing imprecise locations (i.e. multipoints) influences the choice of storing vague spatial information and conventional attributes jointly or in different dimension tables of non-redundant GDW.

The remainder of this paper is organized as follows. Section 2 lists the basic concepts used throughout the paper. Section 3 surveys related works. Section 4 investigates the performance benefits of the joint storage of vague spatial objects in non-redundant GDW schemas. Section 5 gives experimental results for vague spatial objects using query windows with uncertainty. Section 6 concludes the paper.

2 Theoretical Foundation

This section lists the basic concepts used throughout the paper.

2.1 The Egg-Yolk model

In the Region Connection Calculus (RCC) (Randell and Cohn 1989), the regions are seen as the main elements and may be of any dimension, but

they all must be of the same dimensionality and must be spatially extended regions as well. This calculus is based on a primitive relation denoted by *C(x,y)* indicating that *x* connects *y*. This primitive relation is reflexive and symmetric and holds when the distance between the two regions *x* and *y* is zero. Any degree of connection between regions is allowed, from external contact to identity and based on the primitive relation *C(x,y)*, a set of region-region relations was defined in (Cohn et al., 1996). By using the primitive relation *C(x,y)*, predicates to express the topological shape of certain regions (e.g. a region with a single hole) have also been defined.

In addition, RCC calculus has been extended to show how uncertainty is handled by modeling relations between regions with indeterminate boundaries using the Egg-Yolk model (Cohn and Gotts 1996). This model defines the Egg as the maximal extent of a vague region and the yolk its minimal extent. By using a set of five RCC relations (called RCC5) to relate eggs and yolks and assuming that yolks are never null, Cohn and Gotts identified forty-six egg-yolk relations between two eggs and grouped different configurations according to the relations involving the pairs of egg-egg, yolk-yolk, and yolk-egg. Figure 1 exemplifies the Egg-Yolk model and considers three points: one point that is certainly inside the region, one point that may or may not be inside the vague region, and one point that is surely outside the region. In Figure 1, the Yolk is dark gray colored, while the light gray region delimits the Egg.

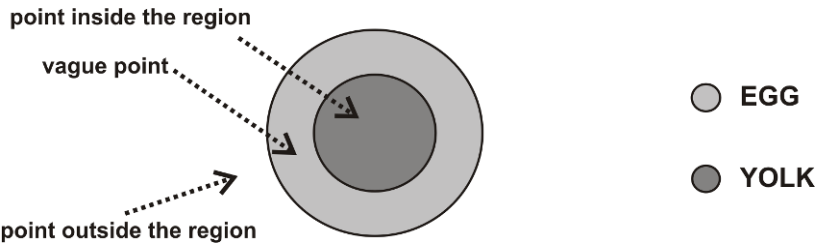


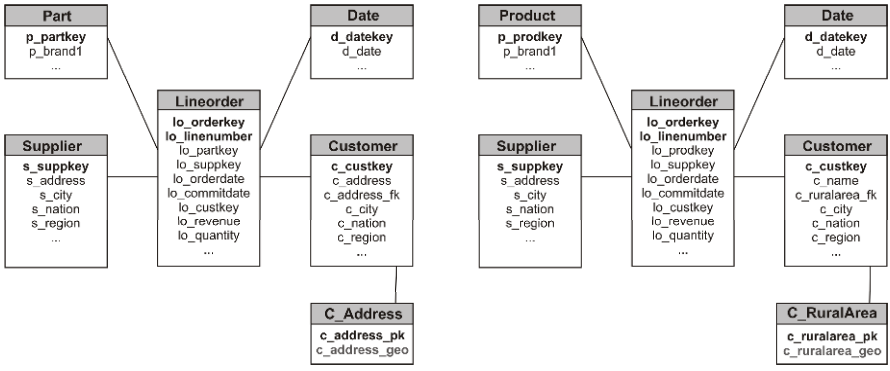
Fig. 1. The Egg-Yolk model.

2.2 Geographic data warehouse

Star schemas having fact and dimension tables are often used in conventional DW applications (Kimball and Ross 2002). Fact tables store numeric measures that indicate the scores of business activities, while dimension tables have attributes that describe and group the values of these measures. In GDW, spatial data are stored as specific attributes of dimension tables or as spatial measures in fact tables (Malinowski and Zimányi 2008; Times

et al., 2008). A star schema can be used to organize data in a GDW and has been classified according to the following types: redundant GDW schema and non-redundant GDW schema. In a redundant GDW schema, the dimension tables store both conventional and spatial attributes and, as a result, repeated spatial objects are maintained. For instance, Utrecht's map is stored in every row whose supplier is located in Utrecht. On the other hand, the non-redundant GDW schema defines two types of dimension tables. The spatial dimension table stores the ID and the vector geometry of each spatial object, while the corresponding conventional dimension table contains the conventional attributes and a foreign key to the spatial dimension table. Previous works identified that spatial data redundancy is associated to greater storage requirements and low query processing performance (Siqueira et al. 2009; Mateus et al., 2010). Therefore, redundant GDW schemas were not considered in this paper.

In this paper, a spatial attribute is seen as a vague spatial attribute once it may contain a set of points indicating uncertain locations or a pair of concentric regions used to represent undetermined boundaries of a given spatial object, according to the Egg-Yolk representation model. In this paper, we focus on Spatial OLAP (SOLAP) queries based on the lowest spatial granularity level of non-redundant GDW schemas, since this is the level with the highest cardinality. Also, this level associates the IDs and the spatial objects through a 1:1 relationship.



(a) GDW with uncertain point locations (b) GDW with vague regions

Fig. 2. Example of non-redundant GDW schemas

For instance, Figures 2a and 2b show non-redundant GDW schemas, where spatial data related to customers are stored in the spatial dimension tables *C_Address* and *C_RuralArea*. These are referenced by foreign keys

(with suffix *_fk*). Both schemas were adapted from the Star Schema Benchmark (O'Neil et al. 2009) and summarize the original hybrid schema in Siqueira et al (2009. They are also given as our running examples of vague GDW schemas. The uncertain customer location is given by a multipoint geometry in attribute *c_address_geo*, while *c_ruralarea_geo* is the vague region with undetermined boundaries of a rural area maintained by the customer, over which a certain agricultural product is administered

2.3 The SB-index

The SB-index (Siqueira et. al 2009) has a sequential structure whose entries maintain a primary key value for the spatial dimension table, a minimum bounding rectangle (MBR), and an implicit pointer to a star-join bit-vector. The *i-th* entry of the SB-index points to the *i-th* bit-vector of a star-join Bitmap index (O'Neil and Graefe, 1995). The bit-vectors describe the tuples of the fact table where a specific key value and its corresponding spatial object (represented by its MBR) occur (i.e. bit value 1) and do not occur (i.e. bit value 0). Figure 3 exemplifies the SB-index for the spatial dimension table *C_Address* of Figure 2a. Note that *c_address_pk* = 1 occurs in the first and second tuples of the fact table, as well as the MBR defined by ((84.64 42.58), (79.45 49.27)).

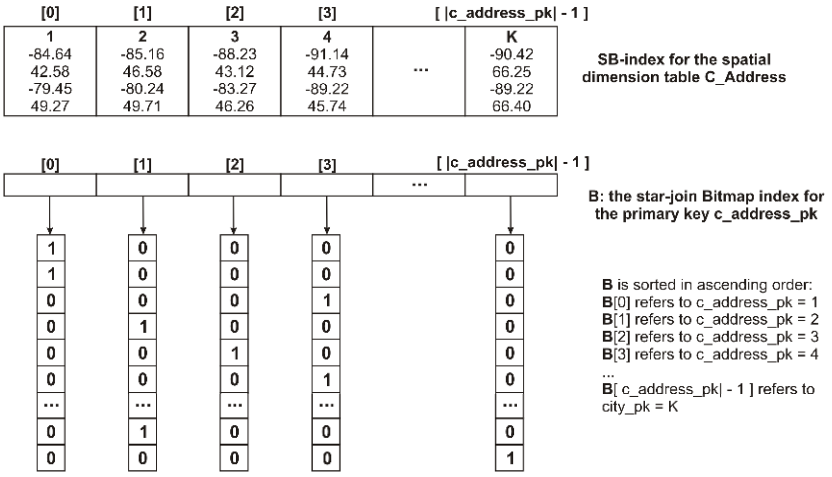


Fig. 3. The SB-index for the spatial dimension table C_Address.

The SB-index query processing can be divided into four phases: (i) the spatial filter step that performs a sequential scan on the index, evaluates the spatial predicate against the MBRs of the entries, and collects key val-

ues considered candidates; (ii) the refinement step that accesses the database to evaluate the spatial predicate against the original objects considered candidates; (iii) the replacement of the spatial predicate by a conventional one, using the key values considered as answers of the spatial predicate, i.e. rewriting the query to contain only conventional predicates; and (iv) accessing the star-join Bitmap index to evaluate the complete query that has only conventional predicates

3 Related work

The motivation behind including uncertain spatial data in GDW is twofold. First, many spatial objects in GIS are inherently vague (e.g. rural areas) or their geometries may be unknown or may have been collected with poor quality. As a result, a number of studies have been carried out using different research methods in order to: (i) determine the kind of spatial relations needed for a GIS from a human cognitive perspective (Freundschuh, 1992); (ii) elaborate formal theories of spatial representation and reasoning (Egenhofer and Franzosa, 1991); (iii) analyze formal models to investigate whether they match human perception and thinking about space (Mark and Egenhofer 1994); (iv) develop automated systems to solve spatial reasoning problems qualitatively (Sharma et al., 1994); (v) investigate the performance effects of using qualitative spatial data to solve GIS users' queries (Oman 1996), and (vi) apply fuzzy logic and fuzzy sets to handle and index spatial objects (Somodevilla and Petry, 2004; Petry, Ladner and Somodevilla, 2007), as well as to define spatial operators (Dilo, By and Stein, 2007) and spatial relationships (Cobb, Petry and Shaw, 2000). Secondly, many questions in management decisions do not require crisp answers or are mostly based on qualitative information. This resulted in the development of approaches for: (i) incorporating vagueness concepts into conventional DW (Fasel and Shahzad 2010); (ii) using membership functions to speed up the Extraction, Transformation and Load (ETL) process of GDW (David et al., 2007); and (iii) designing a methodology to help in the construction of fuzzy DW (Sapir et al., 2008).

However, there may be a lack in literature concerning quantitative studies to assess the costs of handling vague spatial information in GDW. On the other hand, several techniques have been proposed to improve query processing performance over crisp GDW. They can be classified according to the following groups: (i) the use of materialized views (Rao et al., 2004); (ii) the horizontal or vertical data fragmentation in one site or several sites in a distributed environment (Ciferri et al., 2007); (iii) the parti-

tion of data across multiple processors to enable parallel processing (Furtado 2004); (iv) the use of index structures (Siqueira et al., 2009); and (v) the design of efficient data schemas to reduce query response times and minimize data storage costs (Siqueira et al., 2009; Mateus et al. 2010). Nevertheless, these works do not focus on the representation and computation of spatial vagueness.

Concerning indexing, the SB-index was proposed in (Siqueira et al., 2009) for improving query performance on redundant and non-redundant GDW. Comparisons of the SB-index approach, the star-join aided by R-tree, and the star-join aided by GiST indicated that the SB-index significantly improves the elapsed time in query processing from 25% up to 99% with regard to spatial predicates of intersection, enclosure, and containment found in roll-up and drill-down operations. However, the SB-index was not applied to vague spatial data with higher cardinality attributes. In addition, Kalashnikov et al. (2006), Li et al. (2007), and Yuen et al. (2010) introduced indices and evaluated their query processing performance for retrieving vague spatial data, but these data were not stored in multidimensional structures such as GDW.

Regarding the design of data schemas, Mateus et al. (2010) examined the impact of using crisp points in the lower level of granularity, which contains the greatest amount of spatial objects. For that study, the spatial predicate containment was used together with spatial query windows that provided low selectivity to the referred spatial predicate due to the windows reduced size (i.e. a small percentage of the extent). Differently from that work, the current paper evaluates vague spatial objects as to imprecise locations using multipoints.

Also, differently from existing works, this paper investigates the effects of storing and processing spatial vague information in GDW without spatial redundancy, and according to different types of vague spatial objects, such as polygons based on the Egg-Yolk model and multipoints. An experimental evaluation on storing multipoints jointly with or separated from conventional dimension tables is presented. In addition, we evaluate the SB-index when dealing with spatial vagueness and high cardinality.

4 Experimental Evaluation

Our performance evaluation addresses current technologies of spatial database systems for dealing with vague spatial data that are stored in non-redundant GDW and have different spatial representations. We also take

into account the indexing issue aiming at improving query processing performance over a non-redundant GDW containing vague spatial data.

4.1 Workbench and workload

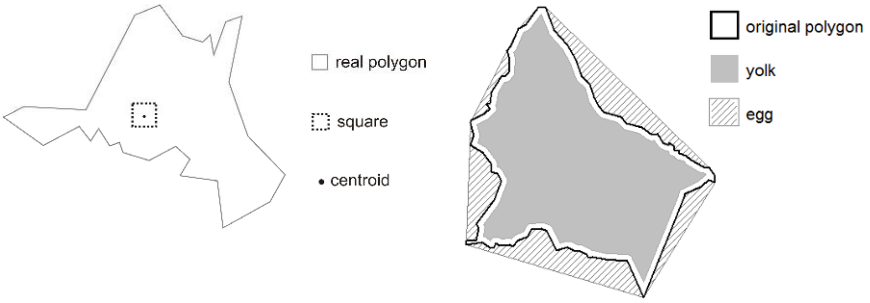
The GDW schemas depicted in Figure 2 were built using the Star Schema Benchmark scale factor 10, generating 60 million facts and 300,000 customers. We defined the workload adapting the Query Q2.3 of that benchmark as below, since it provides all GDW query characteristics: joins among huge tables, spatial and conventional predicates, aggregation and sorting. The spatial predicate is based on the *SpatialRelationship*, which evaluates a high cardinality spatial attribute, i.e. *c_address_geo* or *c_ruralarea_geo* both with a cardinality of 300,000, against a rectangle *QW* that was not previously stored in any dimension table.

```
SELECT SUM (lo_revenue), d_year, p_brand1
FROM lineorder, date, part, customer, c_address
WHERE lo_orderdate = d_datekey  AND lo_partkey = p_partkey
 AND lo_custkey = c_custkey AND c_address_fk = c_address_pk
 AND p_brand1 = 'MFGR#2239'
 AND SpatialRelationship(spatial_attribute, QW)
GROUP BY d_year, p_brand1
ORDER BY d_year, p_brand1;
```

As already stated, we intended to investigate the use of different representation types of uncertain spatial data in the spatial attribute. Firstly, we gathered real polygons from the rural census of the Brazilian Institute of Geography and Statistics (http://www.ibge.gov.br). We employed multipoints to express the imprecise location of a point object. We generated multipoints composed of 4 up to 24 points as shown in Figure 4a. Firstly, the centroid of the real polygon was obtained. Then, a square was temporarily built around the centroid. Finally, the set of points was placed on the sides of the square. After placing the points, the square was deleted. This procedure was applied to all real polygons. All generated multipoints were stored in the *c_address_geo* attribute.

We have also adopted the real polygons to create vague regions according to the Egg-Yolk model, as shown in Figure 4b. In order to build these regions, the Yolk was obtained by applying a negative buffer on the real polygon, while the Egg was designed by calculating the convex hull of the real polygon. As a result, the Egg part of the vague object has much less points than the Yolk part. These complex polygons were used to represent regions with a known location, but with undetermined boundaries. The use of the convex hull benefited the spatial database routines to determine a given intersection relationship and represents a worst-case scenario when

compared to polygons composed of a random geometry with high complexity (i.e. a high number of vertices). These vague regions were stored in the *c_ruralarea_geo* attribute.



a) Creating a multipoint b) Creating an Egg-Yolk polygon

Fig. 4. Examples of the generated vague spatial data.

The performance tests were carried out on a computer with a 3.2 GHz Pentium D processor, 8 GB of main memory, a 7200 RPM SATA 750 GB hard disk with 32 MB of cache, Linux CentOS 5.2, PostgreSQL 8.2.5, PostGIS 1.3.3, and FastBit version 0.9.2b. The SB-index was implemented using the C/C++ programming language and the disk page size was set to 4 KB. We employed FastBit version 0.9.2b as the Bitmap software. When evaluating the performance of the spatial database resources, GiST indices were built on the spatial attributes.

4.2 Querying uncertain locations

Given a bi-dimensional rectangle *R* whose sides are parallel to the axes of their respective dimensions, called **crisp query window**, the range query finds all the objects that satisfy a given topological relationship with respect to *R*. Each type of topological relationship (e.g. intersects, within) characterizes a specific subtype of a range query. Despite our dataset maintains vague spatial data, the aforementioned range query precisely retrieves the objects that satisfy the relationship based on a crisp query window. Figure 5 depicts the multipoints and the query window.

The first set of experiments used multipoints to represent addresses with vague location and was applied only over spatial database resources aided by spatial indices. We defined two GDW schemas: the first is the schema described in Figure 2a, while the second one was designed as an adaption of it, in which we added the *c_address_geo* attribute in the *Customer* dimension table and dropped the table *C_Address*. There is a 1:1 association

between *c_address_geo* and *c_custkey*, as a result, this schema adaption avoids unnecessary joins and store conventional and spatial data jointly. The containment range query was the query's spatial predicate.

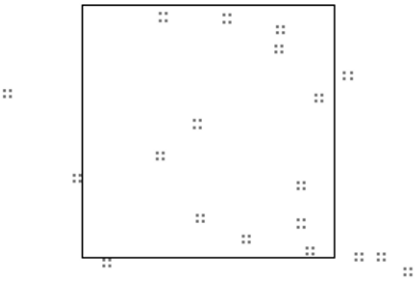


Fig. 5. Query Window for multipoints composed of 4 points.

We performed 5 consecutive queries (described in Section 4.1) using disjoint spatial query windows which covered each one 0.10% of the extent, and gathered the average elapsed time. Table 1 shows the performance results. The joint storage of conventional data and vague spatial objects in dimension tables produced a performance improvement up to nearly 10% when compared with their separate storage, considering a few points (e.g. 4). This performance gain is very similar to the findings of Mateus et al. (2010) for crisp spatial locations. However, as the number of points found in the multipoint feature increased, the performance gain decreased, i.e. for 24 points there is a performance loss of 7% approximately. This pattern prevents the joint storage of multipoints in dimension tables for a higher number of points. Therefore, as many points represent a multipoint, the separate storage of vague spatial data and conventional data is recommended and as a result the additional join costs do not impair the query processing performance.

Table 1. Performance results to process intersection range queries using crisp query windows and multipoints as vague objects in a GDW.

Points	Joint storage	Separate storage	Time Reduction (%)	Customer table	C_Address table
4	56.14 sec.	63.33 sec.	10.18	101 MB	55 MB
12	61.71 sec.	67.09 sec.	8.36	141 MB	101 MB
24	62.19 sec.	58.25 sec.	-6.77	248 MB	161 MB

Regarding the storage costs, Table 1 shows that as the number of points increases, the table size is larger. Maintaining conventional and spatial attributes jointly (column *Customer table*) is more costly than keeping a separate table to store the multipoints (column *C_Address table*). The lar-

ger table size impairs the query processing performance when using the joint storage of both spatial and conventional data.

4.3 Querying regions with undetermined boundaries

The second set of experiments used the polygons built according to the Egg-Yolk model to represent vague addresses, and assessed the intersection range query using disjoint query windows, as shown in Figure 6. We also gathered the average of the elapsed time for 5 consecutive queries. The GDW schema used was the one depicted in Figure 2b and the vague polygons were stored in the *c_ruralarea_geo* attribute.

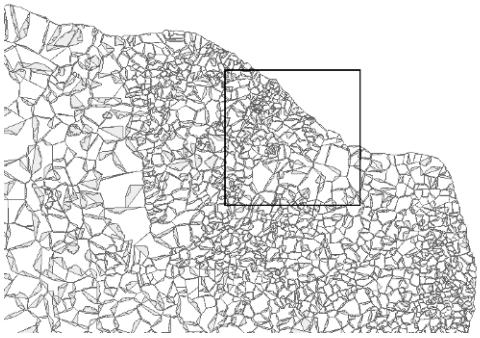


Fig. 6. Intersection range query using polygons based on the Egg-Yolk model

Concerning the storage requirements, the *C_RuralArea* table occupies 3,328 MB. This is a very high cost if compared to tables with geometries with less complexity, like those mentioned in Section 4.2. With respect to query processing performance, Table 2 lists the elapsed times in seconds of query processing over both the SB-index and the DBMS resources, according to increasing query window sizes. The column *QW/Extent* states how much of the extent's area is covered by the query window, while the column Selectivity shows the selectivity of the spatial predicate. The Time Reduction column calculates how much faster the SB-index was than the DBMS resources.

According to Table 2, the larger the query window, the higher is the selectivity of the spatial predicate, since more polygons satisfy the intersection spatial predicate. The SB-index outperformed the DBMS resources for selectivity up to 0.19%. Outstanding results were observed for very low selectivity values (up to 90.35% for the 0.026% selectivity) while a significant time reduction of 36.94% was provided for the selectivity of 0.19%. On the other hand, higher values of selectivity determined that the DBMS

resources outperformed the SB-index. The fact that the polygon's Egg is a convex hull benefited the DBMS query processing as explained in Section 3, since this approximation can be used to efficiently filter the spatial objects. However, the SB-index does not manipulate convex hull, therefore its filter step based on MBRs introduced several false candidates to be evaluated in a further refinement step. The use of polygons with random shape and high complexity (i.e. number of vertices) could improve the performance gains of the SB-index even more.

We have also observed that, in the worst performance result of the SB-index (last line of Table 2), the refinement step spent 77.39% of the total elapsed time. The filter step consumed only 0.08% of the total elapsed time and the access to the star-join Bitmap index spent 22.53%. This analysis corroborated the need for an intermediate step between the spatial filter and the refinement in the SB-index, especially when adopting a higher spatial predicate selectivity and dealing with vague polygons.

Table 2. Performance results to process intersection range queries using crisp query windows and polygons as vague objects in a GDW.

QW/Extent	Selectivity	SB-index	DBMS	Time Reduction (%)
0.01%	0.0260%	7.05	73.08	90.35
0.10%	0.19%	43.56	69.08	36.94
0.25%	0.40%	88.78	68.23	-30.13
0.50%	0.81%	142.97	73.94	-93.35

4.4 Improving the SB-index's query processing performance

In order to enhance the query processing performance of the SB-index without changing its data structure, some adaptations on its query processing algorithm were proposed. Instead of collecting all candidates to submit them to the refinement step, we determined that when N candidates are collected during the sequential scan, they should be submitted to refinement immediately. After this refinement, the spatial predicate answers are added to a conventional predicate that composes the rewritten query (i.e. a query containing only conventional predicates). This rewritten query is also executed immediately. After that, the sequential scan continues until it collects N candidates and starts the new refinement process again. When the sequential scan finishes, there might be M candidates ($M<N$) that should be passed to the refinement process. After executing all sub-queries, the results of these sub-queries are combined. The idea behind executing several sub-queries instead of a single query is to quickly fetch

previously cached objects in the main memory. However, differently from Section 4.3, in this experiment, complete queries were not executed consecutively, i.e. all cache and buffers were flushed after finishing each complete query (such as that described in Section 4.1).

The described enhancement was applied to the SB-index and the selectivity of 0.81%, as shown in Table 3. Although the time reduction provided by this mechanism was low, it has significantly improved the SB-index performance. However, the low time reduction reinforces the need of an intermediate step between SB-index' filter and refinement phases, using another approximation for the spatial object, such as the convex hull that is used by the spatial database resources.

Table 3. Performance results for the improved query processing of the SB-index.

QW/Extent	Selectivity	SB-index	DBMS	Time Reduction (%)
0.50%	0.81%	98.79	99.23	0.49

5 Processing vague range queries

In this section, we argue that undetermined boundaries are also applied to the bi-dimensional rectangle *R* (i.e. for the query window) of a range query. In this sense, *R* is replaced by a pair of bi-dimensional concentric rectangles to denote an uncertain region as shown in Definition 5.1

Definition 5.1 *(Vague Range Query): VRQ identifies all spatial objects with undetermined locations that satisfy two (or more) topological relationships concerning two (or more) bi-dimensional, iso-oriented and concentric rectangles, so that:*

(i) The first topological relationship is verified according to the inner rectangle of the range query and represents the containment relationship (i.e. CRQ) to be more selective and consequently, denote a lower degree of uncertainty;

(ii) The second relationship is computed using the outer range query rectangle and denotes the intersection relationship (i.e. IRQ) to be less restrictive and as a result, to gradually indicate a greater degree of uncertainty.

In Figure 7, the spatial objects within query window 1 (QW1) provide results with more reliability than those objects that intersect query window 2 (QW2). Clearly, the spatial objects within QW1 intersect QW2. However, the results must be kept separate, since the first relationship has pri-

ority over the second. We designed Figure 7 with crisp polygons for the sake of simplicity, but vague query windows should also be applied over a vague GDW. In our experiments, QW1 covers 0.25% of the extent, while QW2 covers 0.5% of the extent.

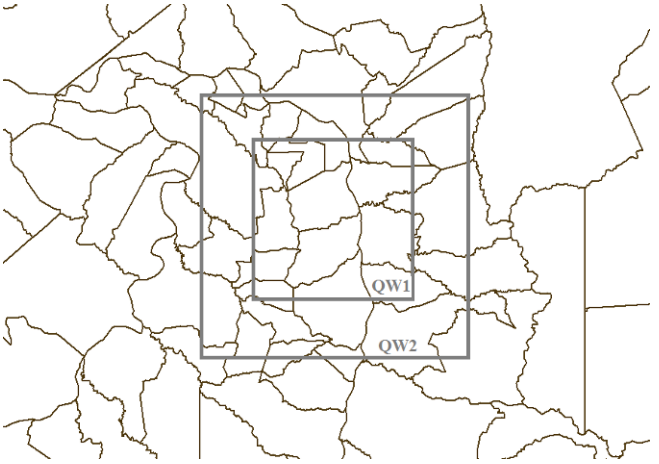


Fig. 7. An example of the use of vague query windows over crisp spatial data.

We had also adapted the SOLAP query described in Section 4.1 to comprise such a pair of query windows, as follows.

```
SELECT SUM (lo_revenue), d_year, p_brand1
FROM lineorder, date, part, customer, c_address
WHERE lo_orderdate = d_datekey  AND lo_partkey = p_partkey
AND lo_custkey = c_custkey  AND c_address_fk = c_address_pk
AND p_brand1 = 'MFGR#2239'
AND WITHIN(c_ruralarea_geo, QW1)
GROUP BY d_year, p_brand1
ORDER BY d_year, p_brand1
UNION
SELECT SUM (lo_revenue), d_year, p_brand1
FROM lineorder, date, part, customer, c_address
WHERE lo_orderdate = d_datekey  AND lo_partkey = p_partkey
AND  lo_custkey  =  customer.c_custkey    AND  c_address_fk  =
c_address_pk
AND p_brand1 = 'MFGR#2239'
AND INTERSECTS(c_ruralarea_geo, QW2)
AND NOT WITHIN(c_ruralarea_geo, QW1)
GROUP BY d_year, p_brand1
ORDER BY d_year, p_brand1;
```

Figure 8 compares the performance results to process a VRQ over the GDW depicted in Figure 2b, using the SB-index and the DBMS. Firstly, it is possible to note that the DBMS configuration outperformed the SB-

index, which introduced an 18.9-second increase over the query processing elapsed time (i.e. almost 14%). Hence, we deeply analyzed the query processing cost of the SB-index to identify bottlenecks. The SB-index' spatial filter task is performed twice: one for QW1 and another for QW2. Together, they represent the undermost fraction of 0.06% of the total elapsed time to process the VRQ. On the other hand, the refinement phase to check which spatial objects are within QW1 took 39.76% of the total elapsed time, while the refinement step to check which spatial objects are intersected by QW2 but are not within QW1 took 41.59% of the total elapsed time. As a result, the refinement step is responsible by 81.35% of the total cost to process the VRQ using SB-index (i.e. 126 seconds, approximately), representing a significant overhead. Finally, the access to the star-join Bitmap index, in order to process the rewritten queries (containing only conventional predicates) for QW1 and QW2, have cost less than 10% each one.

The DBMS efficiently manipulated the two rectangles that represent the vague query window and the vague spatial objects by means of the use of convex hull. These results were very distinct from the comparison of the SB-index with the DBMS using crisp query windows and crisp spatial objects, as observed by Siqueira et al. (2009). For this last scenario, SB-index highly improved the query performance of SOLAP queries. Our present results claim for a new index structure proposal for GDW that store vague spatial objects to enable VRQ.

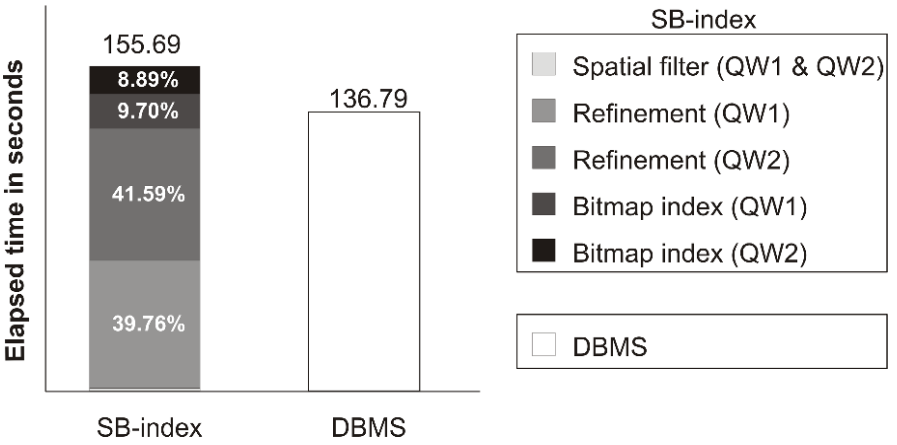


Fig. 8. Performance results to process VRQ.

6 CONCLUSION AND FUTURE WORK

There are several applications that require storage and processing vague spatial data in geographic data warehouses. This paper investigated the handle of vague spatial data stored in dimension tables of a non-redundant geographic data warehouse schema. We analyzed imprecise locations represented as multipoints as well as regions with undetermined boundaries expressed as polygons according to the Egg-Yolk model. We also assessed the processing of crisp and vague spatial query windows in SOLAP queries. Our performance results were gathered for spatial database resources and for the SB-index.

Considering crisp query windows, the joint storage of conventional data and vague spatial objects represented as multipoints in dimension tables produced a performance improvement up to nearly 10% when compared with the separate storage, for multipoints with 4 points. Considering the regions with undetermined boundaries expressed by the polygons, the SB-index outperformed spatial database resources for lower selectivity values of queries up to 90.35%, while for higher selectivity values the SB-index was always outperformed by spatial database resources.

On the other hand, for vague query windows, the SB-index did not improve the query performance when compared with spatial database resources. These results were very distinct from the comparison of the SB-index with spatial database resources using crisp query windows and crisp spatial objects. This fact claims for a new proposal of an index to enhance the query processing performance over GDW that store vague spatial objects. This is one of our future works. We also argue that multidimensional analytical queries enhanced with spatial predicates are essential in geographic data warehouses, such as roll-up, drill-down and drill-across. In this context, we are investigating indexing techniques to support these queries and improve their processing performance. Another future investigation consists of adapting our methods for infinitely valued sets, such as fuzzy sets.

Acknowledgements

This work has been supported by the following Brazilian research agencies: FAPESP, CNPq, CAPES, INEP, and FINEP. The first two authors thank the support of the Web-PIDE Project in the context of the Observatory of the Education of the Brazilian Government. The work carried by the third author was supported by funds from the CNPq under the Grant

479018/2009-0. The last author's work has been funded by FAPESP under the Grant 2009/06052-7.

References

Ciferri, C. D., Ciferri, R. R., Forlani, D. T., Traina, A. J., and Souza, F. F. (2007). Horizontal Fragmentation as a Technique to Improve the Performance of Drill-down and Roll-up Queries. ACM SAC (pp. 494-499).

Cobb, M.A., Petry, F.E., Shaw, K.B. (2000). Fuzzy spatial relationship refinements based on minimum bounding rectangles variations. Fuzzy Sets and Systems, v.113 (1), 111-120.

Cohn, A. G., and Gotts, N. M. (1996). The Egg-yolk Representation of Regions with Indeterminate Boundaries. In: P. A. Burrough, and A. U. Frank, Geographic Objects with Indeterminate Boundaries - GISDATA 2 (pp. 171-187).

David, P., Somodevilla, M. J., and Pineda, I. H. (2007). Fuzzy Spatial Data Warehouse: A Multidimensional Model. 8th Mexican International Conference on Current Trends in Computer Science (pp. 3-9).

Dilo, A., By, R.A., Stein, A. (2007). A system of types and operators for handling vague spatial objects. IJGIS 21(4), 397-426.

Egenhofer, M. J., and Franzosa, R. D. (1991). Point-set Topological Spatial Relations. IJGIS, (5), 161-174.

Fasel, D., and Shahzad, K. (2010). A DataWarehouse Model for Integrating Fuzzy Concepts in Meta Table Structures. 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems, (pp. 100-109).

Ferreira, A. C., Campos, M. L., and Tanaka, A. (2001). An Architecture for Spatial and Dimensional Analysis Integration. World Multiconference on Systemics, Cibernetics and Informatics. Volume XIV - Computer Science and Engineering. Part II.

Freundschuh, S. M. (1992). Is There a Relationship between Spatial Cognition and Environmental Patterns? International Conference GIS (pp. 288-304). Pisa, Italy: Springer-Verlag.

Furtado, P. (2004). Experimental Evidence on Partitioning in Parallel Data Warehouses. 7th ACM DOLAP, (pp. 23-30).

Kalashnikov, D., Ma, Y., Mehrotra, S., Hariharan, R. (2006). Index for fast retrieval of uncertain spatial point data. In: ACM GIS 2006, Arlington, USA, pp. 195-202.

Kimball, R. and Ross, M. (2002) The Data Warehouse Toolkit. Wiley, 2nd ed.

Li, R., Bhanu, B., Ravishankar, C., Kurth, M., and Ni, J. (2007). Uncertain Spatial Data Handling: Modeling, Indexing and Query. Computers and Geosciences, (33), 42–61.

Malinowski, E., and Zimányi, E. (2008). Advanced Data Warehouse Design: From Conventional to Spatial and Temporal. (Data-Centric Systems and Applications): Springer Publishing Company, Inc.

Mark, D. M., and Egenhofer, M. J. (1994). Modeling Spatial Relations between Lines and Regions: Combining Formal Mathematical Models and Human Subject Testing. Cartography and Geographic Information Systems, 21(3), 195-212.

Mateus, R. C., Times, V. C., Siqueira, T. L., Ciferri, R. R., and Ciferri, C. D. (2010). How Does the Spatial Data Redundancy Affect Query Performance in Geographic Data Warehouses?. JIDM, v. 1, pp. 519-534, 2010.

O'Neil, P., Graefe, G. (1995) "Multi-table joins through bitmapped join indices," In: ACM SIGMOD Record, v.24, n.3, pp. 8-11.

O'Neil, P., O'Neil, E., Chen, X., and Revilak, S. (2009). The Star Schema Benchmark and Augmented Fact Table Indexing. TPCTC'2009, pp. 237-252.

Oman, C. (1996). GIS and the Channel Tunnel Rail Link. Institution of Civil Engineers, Geographic Information Systems, (pp. 19-22).

Petry, F.E., Ladner, R. and Somodevilla, M. (2007). Indexing implementation for vague spatial regions with R-trees and Grid Files. A. Morris and S. Kokhan (Eds.) Geographic Uncertainty in Environmental Security. Springer, pp.187-199.

Randell, D. A., and Cohn, A. G. (1989). Modelling Topological and Metrical Properties in Physical Processes. In: H. Levesque, R. Brachmann, and R. Reiter, Principles of Knowledge Representation and Reasoning. pp. 55-66.

Randell, D. A., Cui, Z., and Cohn, A. G. (1992). A Spatial Logic based on Regions and Connection. 3rd International Conference on Principles of Knowledge Representation and Reasoning.

Rao, F., Zhang, L., Yu, X., Li, Y. and Chen, Y. (2003) Spatial hierarchy and OLAP-favored search in spatial data warehouse. 6th ACM DOLAP, pp. 48-55.

Sapir, L., Shmilovici, A., and Rokach, L. (2008). A Methodology for the Design of a Fuzzy Data Warehouse. 4th International IEEE Conference on Intelligent Systems, pp. 2-14 - 2-21.

Sharma, J., Flewelling, D. M., and Egenhofer, M. J. (1994). A Qualitative Spatial Reasoner. 6th International Symposium on Spatial Data Handling, (pp. 665-681). Edinburgh, Scotland.

Siqueira, T. L., Ciferri, R. R., Times, V. C., and Ciferri, C. D. (2009). The Impact of Spatial Data Redundancy on SOLAP Query Performance. JBCS, 15 (2), 19-34.

Somodevilla, M. and Petry, F.E. (2004). Indexing Mechanisms to Query FMBRs. NAFIPS'2004, pp. 198-202.

Times, V. C., Fidalgo, R. N., Fonseca, R., Silva, J., and Oliveira, A. G. (2008). A Metamodel for the Specification of Geographical Data warehouses. Annals of Information Systems, (5), 93-114.

Yuen, S., Tao, Y, Xiao, X., Pei, J. (2010) Superseding Nearest Neighbor Search on Uncertain Spatial Databases. IEEE TKDE, v. 22, n. 7, pp. 1041-1055.

A Scalable Approach for Generalization of Land Cover Data

Frank Thiemann[1], Hendrik Warneke[2], Monika Sester[1], Udo Lipeck[2]

[1]Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Hannover, Germany
[2]Institute of Practical Computer Science, Leibniz Universität Hannover, Hannover, Germany
{frank.thiemann, monika.sester}@ikg.uni-hannover.de ,
{warneke, lipeck}@dbs.uni-hannover.de

Abstract. The paper presents a scalable approach for generalization of large land-cover data sets using partitioning in a spatial database and fast generalization algorithms. In the partitioning step, the data set is split into rectangular overlapping tiles. These are processed independently and then composed into one result. For each tile, semantic and geometric generalization operations are performed to remove features that are too small from the data set. The generalization approach is composed of several steps consisting of topologic cleaning, aggregation, feature partitioning, identification of mixed feature classes to form heterogeneous classes, and simplification of feature outlines.
The workflow will be presented with examples for generating CORINE Land Cover (CLC) features from the high resolution German authoritative land-cover data set of the whole area of Germany (DLM-DE). The results will be discussed in detail, including runtimes as well as dependency of the result on the parameter setting.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_20,

1 Introduction

1.1 Project Background

The European Environment Agency (EEA) collects the Coordinated Information on the European Environment (CORINE) Land Cover (CLC) data set to monitor the land-cover changes in the European Union. The member nations have to deliver this data every few years. Traditionally, this data set was derived from remote sensing data. However, the classification of land-cover from satellite images in shorter time intervals becomes more cost intensive.
Therefore, the federal mapping agency in Germany (BKG) investigates an approach of deriving the land cover data from topographic information. The BKG collects the digital topographic landscape models (ATKIS Base DLM) from all federal states. The topographic base data contains up-to-date land-use information; the update rate being one year. This data will be transformed to a high resolution land-cover data set called DLM-DE. After this transformation, there are still some differences between DLM-DE and CLC. Table 1 summarizes the main characteristics of the two data sets.

Data set	CORINE LC	DLM-DE LC
Scale	1:100 000	> 1:10 000
Source	satellite images	aerial images, cadastre
min. area size	25 ha	< 1 ha
min. width	100 m	< 10 m
Classes of heterogeneous agricultural cover	4 / 2 relevant	marginal, mostly separated in its homogeneous components

Table 1. Comparison of ATKIS and CLC

1.2 CORINE Land Cover (CLC)

CORINE Land Cover is a polygon data set in the form of a planar partitioning (or tessellation); polygons do not overlap and cover the whole area without gaps. The scale is 1:100 000. Each polygon has a minimum area of 25 hectares and a minimum width of 100 meters. There are no adjacent polygons with the same land-cover class as these have to be merged.

Land cover is classified hierarchically into 46 classes in three levels, for which a three digit numerical code is used. The first and second level groups are:

1xx artificial (urban, industrial, mine)
2xx agricultural (arable, permanent, pasture, heterogeneous)
3xx forest and semi-natural (forest, shrub, open)
4xx wetland (inland, coastal)
5xx water (inland, marine)

In CLC there are four aggregated classes for heterogeneous agricultural land-cover. Such areas are composed of small areas of different agricultural land-cover. In Germany, only two of these four classes occur. Class 242 is composed of alternating agricultural covers (classes 2xx). Class 243 is a mixture of agricultural and (semi-) natural areas.

1.3 DLM-DE LC

The land cover (LC) layer of the Digital Landscape Model (DLM) of Germany (DE) is a new product of the BKG. DLM-DE LC is derived by a semantic generalization from the Authoritative Topographic Cartographic Information System (ATKIS) which is Germany's large scale topographic landscape model. After selecting all relevant features from ATKIS, the topological problems like overlaps and gaps are solved automatically using appropriate algorithms. The reclassification to the CLC nomenclature is done using a translation table which takes the ATKIS classes and their attributes into account. In the cases where a unique translation is not possible, a semi-automatic classification from remote sensing data is used. The scale of DLM-DE is approximately 1:10 000. The minimum area for polygons is less than one hectare.

1.4 Automatic derivation of CLC from DLM-DE

The aim of the project is the automated derivation of CLC data from ATKIS. This derivation can be considered as a generalization process, as it requires both thematic selection and reclassification, and geometric operations due to the reduction in scale. Therefore, the whole workflow consists of two main parts. The first part is a model transformation and consists of the extraction, reclassification, and topological correction of the data. The derived model is called DLM-DE LC. The second part, the generalization, which will be described in more detail in this paper, is the aggregation,

classification, and simplification for the smaller scale. For that purpose a sequence of generalization operations is used. The operators are dissolve, aggregate, split, simplify, and a heterogeneous class filter. The program computing the generalization is called CLC-generator.

The classification of agricultural heterogeneous areas to 24x-classes in the case that a special mixture of land-covers occurs is one of the main challenges. The difficulty is to separate these areas from homogeneous as well as from other heterogeneous classes.

1.5 Scalability

Another challenge of the project is the huge amount of data. The DLM-DE LC contains ten million polygons. Each polygon consists, on average, of thirty points, so one has to deal with 300 million points, which is more than a standard PC can store in main memory. While fast algorithms and efficient data structures reduce the required time for the generalization, we have developed a partitioning and composition strategy in order to overcome problems due to memory limitations when processing large datasets. We store the source data for the generalization process in a spatial database system and divide it into smaller partitions, which can efficiently be handled by the CLC-generator on standard computers. The resulting CLC-data sets for the individual tiles are then composed into one data set within the database.

To ensure consistency, i.e. to get identical results from partitioned and unpartitioned execution, some redundancy is added to the partitions in the form of overlapping border regions. This redundancy is removed in the composition phase and geographic objects residing at the border of different partitions are reconciled.

The amount of redundancy added can be controlled by the width of the border regions. As bigger regions cause longer running times of the generalization, we are interested in using values as small as possible while still ensuring consistency. Another parameter influencing performance is the number of partitions. The tiles have to be small enough to avoid memory limitations but a fine-granular partitioning leads to more composition overhead. We present experiments targeted at finding the optimal values for these parameters.

2 Related Work

CORINE Land Cover (Büttner et al. 2006) is being derived by the European States (Geoff et al. 2007). In order to link the topographic database with the land-use data, the Federal Agency of Cartography and Geodesy has developed a mapping table, including transformation rules between CLC and ATKIS objects (Arnold 2009). In this way, the semantic mapping has been established by hand, introducing expert knowledge. There are approaches to automate this process, e.g. Kuhn (2006) or Kavouras and Kokla (2008). Jansen et al. (2008) propose a methodology to integrate land-use data.

As described above, the approach uses different generalization and interpretation steps. The current state of the art in generalization is described in Mackaness et al. (2007). The major generalization step needed for the generalization of land-cover classes is aggregation. The classical approach for area aggregation was given by van Oosterom (1995), the so-called GAP-tree (Generalized Area Partitioning). In a region-growing fashion, areas that are too small are merged with neighboring areas until they satisfy the size constraint. The selection of which neighbor to merge with depends on different criteria, mainly geometric and semantic constraints, e.g. similarity of object classes or length of common boundary. This approach is implemented in different software solutions (e.g. Podrenek 2002). Although the method yields areas of required minimum size, there are some drawbacks; a local determination of the most compatible object class can lead to a high amount of class changes in the whole data set. Also, objects can only survive the generalization process, if they have compatible neighbors. The method by Haunert (2008) is able to overcome these drawbacks. He is also able to introduce additional constraints, e.g. that the form of the resulting objects should be compact. The solution of the problem has been achieved using an exact approach based on mixed-integer programming (Gomory 1958), as well as a heuristic approach using simulated annealing (Kirkpatrick 1983). However, the computational effort for this global optimization approach is very high.

Collapse of polygon features corresponds to the skeleton operation, which can be realized in different ways. A simple method is based on triangulation; another is medial axis or straight skeleton (Haunert and Sester 2008).

The identification of mixed classes is an interpretation problem, while interpretation is predominant in image understanding where the task is to extract meaningful objects from a collection of pixels (Lillesand and Kiefer 1999). Additionally, in GIS-data, interpretation is needed, even when

the geo-data are already interpreted. For example, in our case, although the polygons are semantically annotated with land-cover classes, we are looking for a higher level structure in the data which evolves from a spatial arrangement of polygons. Interpretation can be achieved using pattern recognition and model based approaches (Heinzle and Anders 2007).
Partitioning of spatial data has extensively been investigated in the area of parallel spatial join processing. In Zhou et al. (1998), a framework for partitioning spatial join operations in a parallel computer environment is introduced and the impact of redundancy on performance is studied. Newer work (Meng et al. 2007) presents an improved join method for decomposing spatial data sets in a parallel database system. Spatial joins only need to collect partition-wise results, possibly including duplicate elimination. Our task of generalization, however, needs geometric composition of results and context dependencies have to be observed.

3 Generalization Approach

3.1 Data and index structures

An acceptable run time for the generalization of ten million polygons can only be reached with efficient algorithms and data structures. For topology depending operations a topologic data structure is essential. For spatial searching, a spatial index structure is needed; furthermore, structures for one-dimensional indexing are used.

In the project, we use an extended Doubly Connected Edge List (DCEL) as a topologic structure. A simple regular grid (two-dimensional hashing) is used as a spatial index for nodes, edges, and faces. For the DLM-DE, a grid width of 100 meters for points and edges (<10 features per cell) and 1000 meters for faces (40 faces per cell) leads to nearly optimal speed.

3.2 Topological cleaning

Before starting the generalization process, the data has to be imported into the topological structure. In this step, we also look for topological or semantic errors. Each polygon is checked for a valid CLC class. Small sliver polygons with a size under a threshold of e.g. 1 m² will be rejected. A snapping with a distance of 1 cm is done for each inserted point. With a point in polygon test and a test for segment intersection, overlapping poly-

gons are detected and also rejected. Holes in the tessellation can be easily found by building loops of the half-edges which do not belong to any face. Loops with a positive orientation are holes in the data set.

3.3 Generalization operators

Dissolve

The dissolve operator merges adjacent faces of the same class. For this purpose, the edges which separate such faces will be removed and new loops are built.

Aggregate

The aggregation step aims at guaranteeing the minimum size of all faces. The aggregation operator in our case uses the simple greedy algorithm described by van Oosterom (1995). It starts with the smallest face and merges it to a compatible neighbor. This fast algorithm is able to process the data set sequentially. There are different options to determine compatible neighbors. The criterion can be:

- the semantic compatibility (semantic distance),
- the geometric compactness, or
- a combination of both.

CLC	111	112	121	122	123	124	131	132	133	141	142	211	212	213	221	222	223	231	241	242	243	244
111	0	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	2	3	4
112	1	0	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	2	3	4
121	3	3	0	1	1	1	2	2	2	4	4	6	6	6	6	6	6	6	5	5	6	7
122	2	2	1	0	1	1	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4
123	3	3	1	1	0	1	2	2	2	4	4	5	5	5	5	5	5	5	5	5	5	5
124	3	3	1	1	1	0	4	4	4	2	2	6	6	6	6	6	6	5	6	5	6	6
131	3	3	2	2	3	3	0	1	1	4	4	7	7	7	7	7	7	7	7	7	7	7
132	3	3	2	2	3	3	3	0	1	4	4	7	7	7	7	7	7	7	7	7	7	7
133	1	1	1	1	1	1	2	2	0	2	2	3	3	3	3	3	3	3	3	3	3	3
141	3	2	3	3	3	3	3	3	3	0	1	7	7	7	7	7	7	7	7	7	5	5
142	3	2	3	3	3	3	3	3	3	1	0	5	5	5	5	5	5	5	5	5	5	5
211	5	5	5	5	5	5	5	5	5	5	5	0	1	1	4	4	4	3	2	2	2	2

Fig. 1. Small extract of the CLC priority matrix

The semantically nearest partner can be found using a priority matrix. We use the matrix from the CLC technical guide (Bossard et al. 2000) (Figure 1). The priority values are from an ordinal scale, so their differences and

their values in different lines should not be compared. The matrix is not symmetric as there may be different ranks when going from one object to another than vice versa (e.g. settlement → vegetation). Priority value zero is used if both faces have the same class. The higher the priority value, the higher is the semantic distance. Therefore the neighbor with the lowest priority value is chosen.

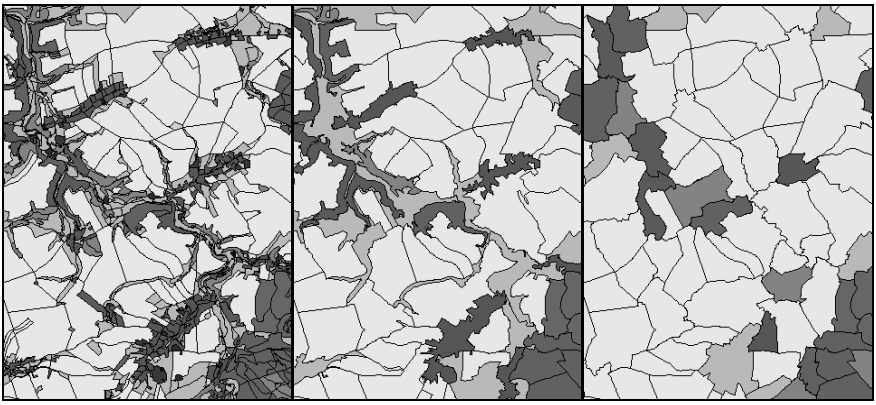


Fig. 2. (Left to right) Original situation, the result of the semantic and geometric aggregation.

As geometric criterion, the length of the common edge is used. A shorter perimeter leads to better compactness. So the maximum edge length has to be reduced to achieve a better compactness.

The effects of using the criteria separately are shown in a real example in Figure 2. The semantic criterion leads to non-compact forms, whereas the geometric criterion is more compact but leads to a large amount of class change. The combination of both criteria allows merging of semantically more distant objects, if the resulting form is more compact. This leads to Formula 1.

$$distance\ (A,B)\ =\ \frac{b^{priority}}{lengt\,h} \tag{1}$$

The formula means that a b-times longer shared edge allows a neighbor with the next worse priority. The base b allows weighting between compactness and semantic proximity. A value of b=1 leads to only compact results; a high value of b leads to semantically optimal results. Using the priority values is not quite correct; it is only a simple approximation for the semantic distance.

Another application of the aggregation operation is a special kind of dissolve that stops at a defined area size. It merges small faces of the same class to bigger compact faces using the geometric aggregation with the condition that only adjacent faces of the same class are considered.

Split

In addition to the criterion of minimal area size, the extent of the polygon is limited to a minimum distance. That demands for a collapse operator to remove slim, elongated polygons, and narrow parts. The collapse algorithm by Haunert and Sester (2008) requires buffer and skeleton operations that are time consuming. Therefore, as a faster alternative, a combination of splitting such polygons and merging the resulting parts with a geometric aggregation to other neighbors is used. Instead of shrinking the slim parts to their medial axes, we split it at suited points and use the aggregation step to merge the slim polygons with another neighbor.

To find the narrows we use a constrained Delaunay triangulation of the polygon. Each triangle is checked for edges and heights smaller than a threshold. These edges or heights will be used for splitting (see Figure 3).

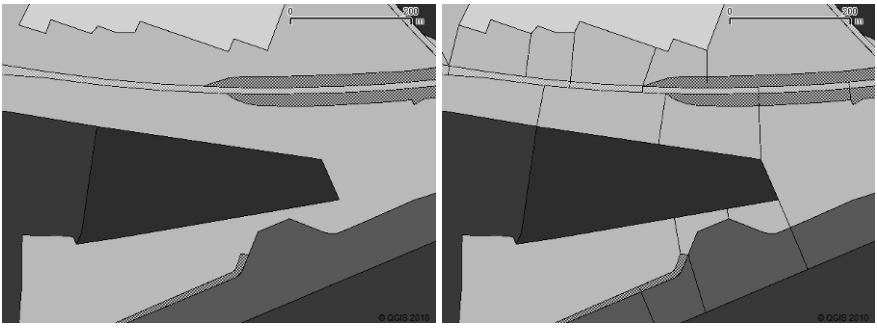


Fig. 3. Data before and after a 100 m split operation.

24x-Filter

In CORINE land-cover, there is a group of classes which stands for heterogeneous land-covers. The classes 242 and 243 are relevant for Germany. Class 242 (complex cultivation pattern) is used for a mixture of small parcels with different cultures. Class 243 is used for land that is principally occupied by agriculture with significant areas of natural vegetation.

Heterogeneous classes are not included in the DLM-DE. To form these 24x-classes, an operator for detecting heterogeneous land-cover is needed. The properties of these classes are that smaller areas with different, mostly agricultural land-cover alternate within the minimum area size (actually 25 ha in CLC). For the recognition of class 242, only the agriculture areas (2xx) are relevant. For 243 also forest, semi- and natural areas (3xx, 4xx) and lakes (512) have to be taken into account.

The algorithm calculates some neighborhood statistics for each face. All adjacent faces within a distance of the centroid smaller than a given radius and with an area size smaller than the target size are collected by a deep search in the topological structure. The fraction of the area of the majority class and the summarized fractions of agricultural areas (2xx) and (semi-) natural areas (3xx, 4xx, 512) are calculated. In the case that the majority class dominates (>75 %), then the majority class becomes the new class of the polygon. Otherwise, there is a check if it is a heterogeneous area or only a border region of larger homogeneous areas.

For that purpose the length of the borders between the relevant classes is summarized and weighted with the considered area. A heterogeneous area is characterized by a high border length, as there is a high number of alternating areas. To distinguish between 242 and 243 the percentage of (semi-natural) areas has to be significant (>25 %).

Simplify

The simplify-operator removes redundant points from the loops. A point is redundant if the geometric error without using this point is lower than an epsilon and if the topology does not change. Therefore we implemented the algorithm of Douglas & Peucker (1973) with an extension for closed loops and a topology check.

3.4 Process chain

In this section the use of the introduced operators and their orchestration in the process chain is shown. The workflow for a target size of 25 ha is as follows:

1. import and clean data and fill holes
2. dissolve faces < 25 ha
3. split faces < 100 m
4. aggregate faces < 1 ha geometrically (base 1.2)
5. reclassify faces with 24x-filter (r=282 m)
6. aggregate faces < 5 ha weighted (base 2)

7. aggregate faces < 25 ha semantically
8. simplify polygons (tolerance 20 m)
9. dissolve all

During the import step (1) semantic and topology is checked. Small topologic errors are resolved by snapping. Gaps are filled with dummy objects. These objects will be merged to other objects in the later steps.

A first dissolve step (2) merges all faces with an adjacent face of the same CLC class which are smaller than the target size (25 ha). The dissolve is limited to 25 ha to prevent polygons from being too large (e.g. rivers that may extend over the whole partition). This step leads to many very non-compact polygons. To be able to remove them later, the following split-step (3) cuts them at narrow internal parts (smaller than 100 m). Afterwards an aggregation (4) merges all faces smaller than 1 ha (100 m × 100 m) to geometrically fitting neighbors.

The proximity analysis of the 24x-filter step (5) re-classifies agricultural or natural polygons smaller than 25 ha in the 25 ha (corresponding to a radius of 282 m) surrounding as heterogeneous (24x class).

The next step aggregates all polygons to the target size of 25 ha. First, we start with a geometric/semantically weighted aggregation (6) to get more compact forms. Second, only the semantic criterion is used (7) to prevent large semantic changes of large areas.

The simplify step (8) smoothes the polygon outlines by reducing the number of nodes. As geometric error tolerance 20 m (0.2 mm in the map) is used. The finishing dissolve step (9) removes all remaining edges between faces of same class.

4 Partitioning method

To create an appropriate partitioning for the derivation of land-cover data from the DLM-DE, we divide the minimum bounding rectangle (mbr) of the data set into a grid of same-sized rectangles. The number of partitions is specified for the x- and y-dimension, respectively. Alternatively, a predefined partitioning scheme, whose geometries have to be imported into the database before processing the topographic data, can be used and are intersected with the data set mbr to identify the relevant partitions.

For each tile three phases are executed called the partitioning, generalization, and composition phase. The second phase is described in detail in the previous section. Below we present the details of the first and the third phase.

The partitioning is defined on the mbr instead of the exact shape of the whole data set for performance reasons. Empty partitions resulting from a non-rectangular shape of the data set are identified in the partitioning phase and skipped.

4.1 Partitioning phase

For the current partition to be processed, first its rectangular geometry is enlarged by adding a user-defined width to its borders. In the following, the area defined by the original rectangle is called interior, the area defined by the enlargement is called border region, and the complement of the enlarged rectangle is called exterior (Figure 4). Then all DLM-objects intersecting the enlarged rectangle are selected and clipped at its border. The resulting data set is exported from the database to be processed by the CLC-generator.

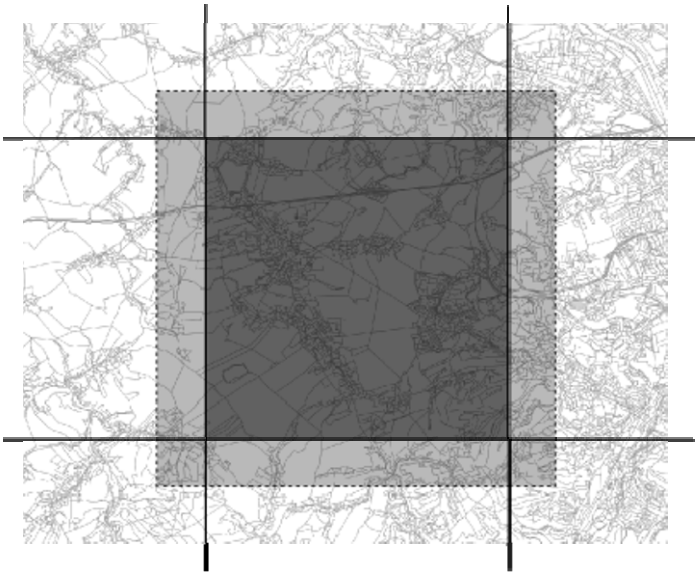


Fig. 4. Partitioning grid with interior (dark grey), border region (light grey), and exterior (white) of one partition.

Please note that each geographic object in a border region also resides in the interior of another partition, which means that these objects are exported and generalized more than once in different partitions. The idea of our partitioning method is that for each area in the interior of the exported partition, enough context is provided to take a correct generalization deci-

sion using only data exported. Areas residing in the exterior are considered too far away to influence the generalization in the interior. Possibly wrong decisions in the border region are removed during composition.

4.2 Composition phase

The result of the generalization phase, which is a valid CLC data set for the current partition, is then imported back into the database. We expect that areas in the border region may be generalized incorrectly because of missing context information in the generalization phase. So the results from the whole border region are thrown away by only selecting areas residing in the interior, clipping these areas at the border of the interior and, adding them to the CLC objects of already finished partitions. Thus the final result will not contain any gap, since each area in the border region also resides in the interior of another partition and is accepted when that partition is processed.

However, because CLC objects are clipped, they do not extend across partition borders, which means that adjacent areas from different partitions but assigned the same land-cover information are represented by two or more polygons. We identify these situations by executing a spatial join searching for objects from the current partition and from already composed neighboring partitions that have the same CLC class and that have a piece of the partition border in common. Reconciliation is done by aggregating (dissolving) each group of objects, which are in this way associated, into one object (see Figure 5).

Fig. 5. Clipped objects from neighboring partitions (left) and reconciled objects (right).

4.3 Implementation issues

We have implemented the partitioning and composition phase on top of an Oracle 11g database with Spatial Data Option installed. All operations accessing data from more than one partition are executed using SQL-statements, so that the database system manages the computation resources and we don't need to deal with memory limitations by ourselves. To avoid unnecessary but expensive disk fetches and geometrical computations, we use Oracle's built-in spatial index type, which is an implementation of the R*-tree (Beckmann et al., 1990), to access the imported DLM data and generated CLC data. To exchange data with the CLC-generator, we use the simple and popular shapefile format by ESRI.

5 Results

5.1 Runtime and memory use of the generalization step

The implemented algorithms are very fast but require a lot of memory. Data and index structures need up to 160 Bytes per point on a 32 bit machine.

The run-time of the generalization routines was tested with a 32 bit 2.66 GHz Intel Core 2 processor with a balanced system of RAM, hard disk, and processor (windows performance index 5.5). The whole generalization sequence for a 45 km × 45 km data set takes less than two minutes. The most time-expensive parts of the process are the I/O-operations which take more than 75 % of the computing time. We are able to read 100 000 points per second from shapefiles while building the topology. The time of the writing process depends on the disk cache. In the worst case, it is the same as for reading.

5.2 Semantic and geometric correctness

To evaluate the semantic and geometric correctness, we did some statistics comparing input, result, and a CLC 2006 reference data set, which was derived from remote sensing data.

Figure 6 shows the input data (DLM-DE), our result, and the CLC 2006 of the test area of Dresden. The statistics in Figure 7 verifies that our result matches with DLM-DE (75 % of the area) better than the reference data set (60 %). This is not surprising as the CLC 2006 used different data sources.

Because of the removing of the small faces our generalization result is a bit more similar to CLC 2006 (66 %) than CLC 2006 to the input dataset.

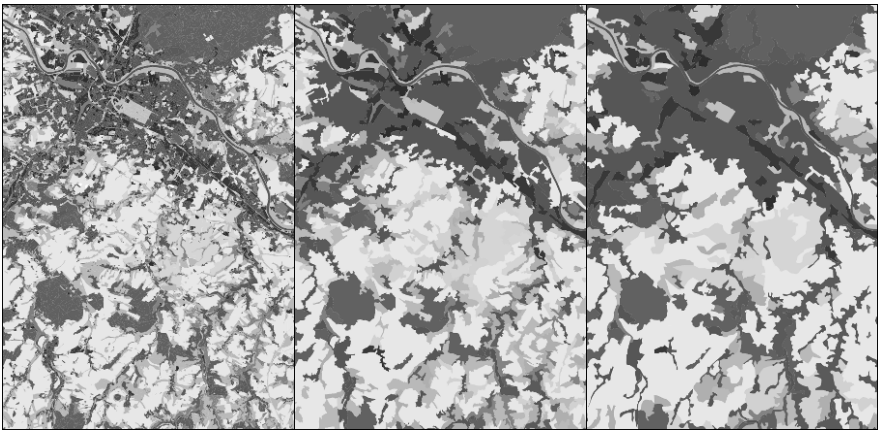


Fig. 6. Extract (20 km × 25 km) of test data set of Dresden from left to right: input DLM-DE, our result and CLC 2006 as reference.

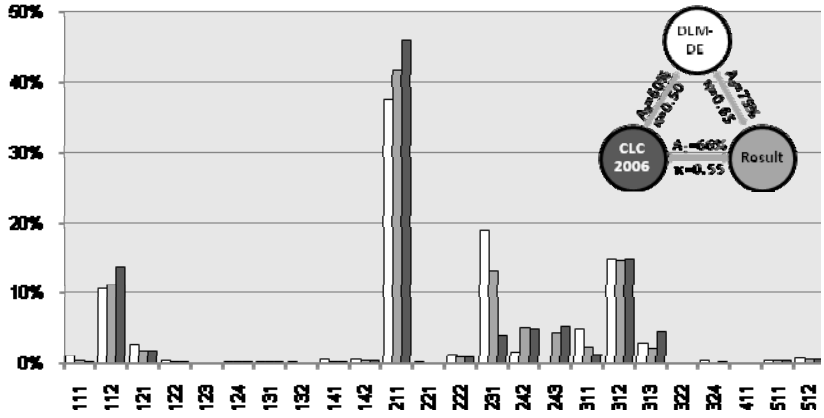


Fig. 7. Percentage of area for each CLC class (bars) and percentage of matching area ($A_{0,}$ area with the same class) and κ-values for the Dresden dataset.

Table 2 shows that our polygons are only a bit smaller, more complex and less compact than the CLC 2006 polygons. The structure index values (diversity, dominance and homogeneity) (Liu et al. 2010) indicate that the structure was preserved during the process.

The percentage of the CLC classes is similar in all data sets (Figure 7). There are some significant differences between the DLM-DE and CLC

2006 within the classes 211/234 (arable/grass land), 311/313 (broad-leaved/mixed forest), and 111/112 (continuous/discontinuous urban fabric). We assume that this comes from different interpretations and different underlying data sources. The percentages in our generated data set are mostly in the middle. The heterogeneous classes 242 and 243 are only marginally included in the input data. Our generalization generates a similar fraction of these classes. However, the automatically generated areas are often not at the same location as in the manually generated reference data set. We argue, though, that this is the result of an interpretation process, where different human interpreters would also yield slightly different results.

Table 2. Statistic of the test data set of Dresden (45 km × 45 km)

Data set	DLM-DE	Result	CLC 2006
Polygons	91717	1244	876
Points per Polygon	23	62	75
Area per Polygon	2.3 ha	167 ha	238 ha
Perimeter per Polygon	0.6 km	10.1 km	9.0 km
Avg. Compactness	50 %	29 %	33 %
Diversity	2.8	2.7	2.6
Dominance	1.9	1.7	1.9
Homogeneity	0.60	0.61	0.57

The input (DLM-DE) and the result match with 75 %. This means that 25 % of the area changes its class during the generalization process. This is not an error; it is an unavoidable effect of the generalization. The κ-values 0.5-0.65, which stand for a moderate up to substantial agreement, should also not be interpreted as bad results, because it is not a comparison with the real truth or with a defined valid generalization.

5.3 Stability of generalization results

To test the influence of the generalization parameters to the result, we made some experiments with our test data sets. To get an impression of its influence and to optimize the generalization, we changed each parameter separately in small steps. The result of the changed generalization was then compared with the input data and the CLC reference data set. The statistics (Table 2) were also taken into account.

To simulate an update process and its effects on the generalized data, we used two different versions of the DLM-DE (a test version with data from 2006 and a refined version with data from 2009) (see Figure 8). The land-

cover of these two data sets differs in nine percent of the area (ground truth). Both data sets were generalized with the same parameters; the land-cover of the generalization results differs in 13 % of the area. 20 % of these differences in generalized data are correct and 20 % are false (different classes). The other 60 % are false positive – they occur at areas where no differences are in ground truth. 30 % of the real changes are missing (false negative) (see Figure 9).

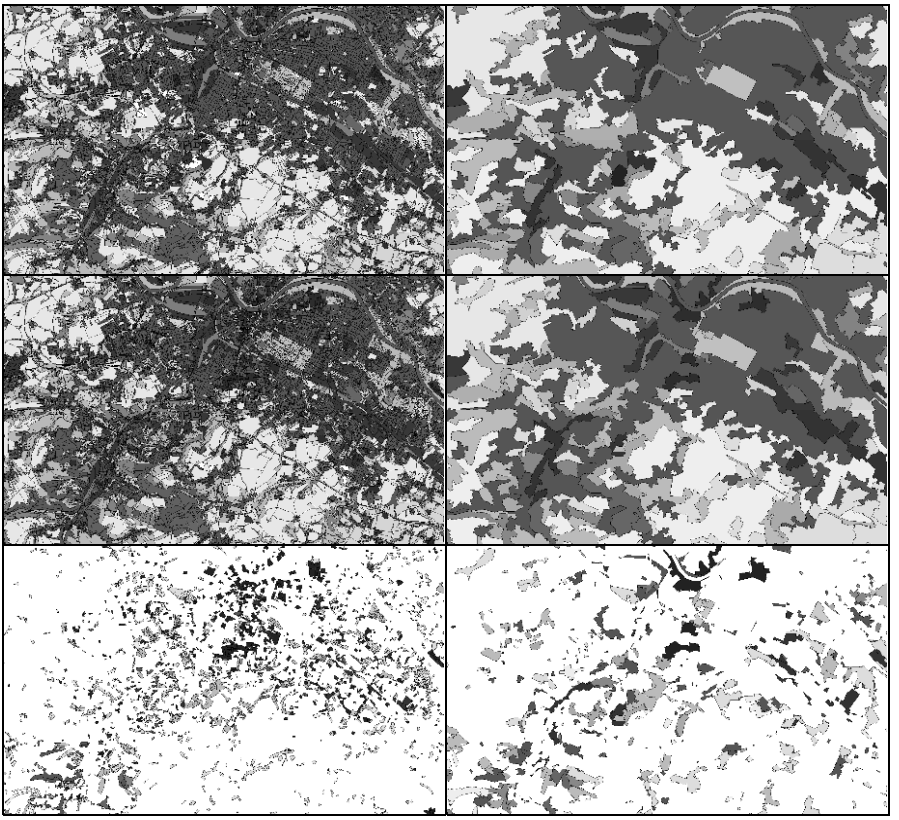


Fig. 8. Two versions of input DLM-DE (left), their generalization results (right) and the differences between the versions (below). 9 % changes of the input data produce 13 % differences in the generalized data.

This example shows that changes in the input data produce more and different changes in the generalized data. The causes of these changes are the classification and the aggregation step. In these generalization operations, decisions are made based on thresholds. A small change can switch between the states under or over the threshold and produce a very different

result. Because of the local decisions of the generalization algorithms, this often leads to changes in the local environment. Changes of the input data have only an influence in a limited environment.

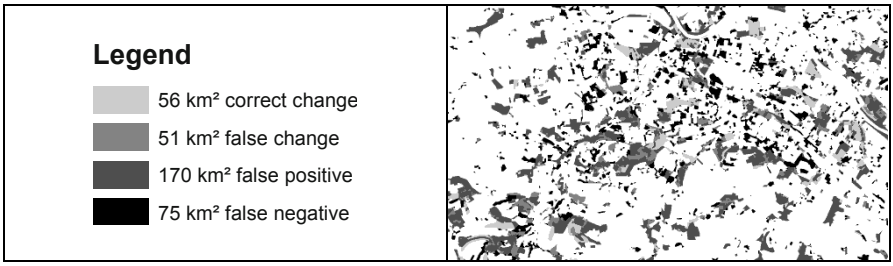


Fig. 9. Overlay of the differences between input and output data.

5.4 Partitioning experiments

To study the impact of the border-width on consistency, we selected the Dresden data set of 45 km × 45 km. We first generalized the data set using only one partition and no border-regions. We considered the result as a reference, because it cannot contain errors induced by partitioning. Second, we generalized the same data set many times while dividing it into four partitions and varying the border-width from zero up to 3.5 km. Each of the results was compared to the reference by computing a diff-data set showing all areas that are assigned different CLC classes. The results are shown in Figure 10 by plotting the sum of differing areas over the width of the partition borders.

These results show the need of adding redundancy to the partitions. While we have a total of almost 2000 hectares of differently classified areas when using no border regions, this error decreases very fast (please note the logarithmic scale on the vertical axis) with increasing border width. At 2.5 km, the diff contains only 0.16 ha ($8 \cdot 10^{-7}$ of the overall generalized area) and at 3.5 km the result matches the reference completely. The running time of the pure generalization rises from 115 to 157 seconds only.

We can prove the capability of our approach to handle large data sets in another experiment, in which we also investigated the connection between partition size and computation time. We generalized the DLM-DE of Lower Saxony, which contains 1.4 million polygons, many times using a different number of partitions. Given the results of the previous experiment, we selected a constant border width of 2.5 km. The running times of the three phases are shown in Figure 11.

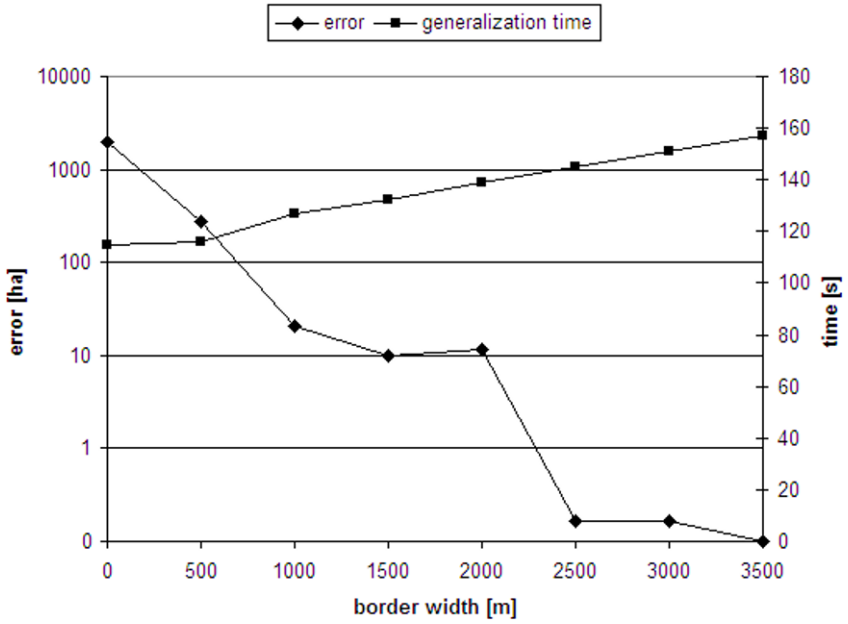


Fig. 10. Total error and running time of the generalization plotted against the border width.

Most noticeable in the experiment is the strong increase of running time for the generalization phase when using large partitions. While it takes only 36 minutes to generalize all partitions of the 5 × 5 grid, using only nine partitions (3 × 3) raises the running time to 90 minutes. Using even larger partitions (e.g. a 2 × 2 grid) is not possible in our test environment due to a lack of free memory available for the CLC-generator. Increasing partition size also has a slightly negative effect on performance in the partitioning phase. Decreasing partition size from 4250 to 1062 km^2 does not alter the running time of any phase significantly. We conclude that partition sizes between 2000 and 4000 km^2 are a good choice for our test environment.

Partitioning was tested on an Intel Core 2 Duo (2.53 GHz) machine with 2 GB RAM running Windows 7. The database server was also locally installed on the same computer.

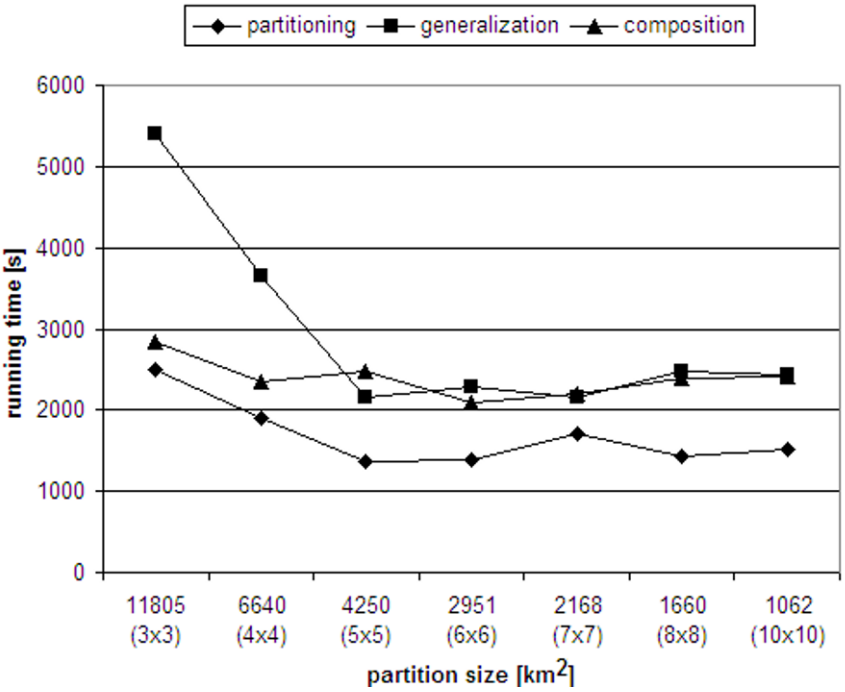


Fig. 11. Total running times of the three phases over all partitions. Number of partitions is given in brackets below the partition size.

6 Conclusions and Outlook

The whole process was designed separating the generalization from scalability issues by only exchanging data in a common file format. This way the CLC-generator could be developed as a stand-alone program without involving a database system but using efficient geometric processing. The database system is only used for partitioning and composition, where data from multiple partitions have to be accessed. Thus, geometric computations that are rather expensive in the database are restricted to intersection (clipping, spatial selections, and joins) and aggregation (reconciling).

We plan to generalize our partitioning concept to a database service that can also be used to solve scalability problems in other localizable computations on large sets of spatial data. We hope this can be done in many practically relevant situations without major changes to the source code of spatial data computations and without major performance overhead caused by partitioning and composition.

Our next project aim is to derive the CORINE land cover change layer from different versions of DLM-DE. The change layer cannot be generated by intersecting CORINE land cover data sets, because the minimum mapping unit of the change layer. This is only five hectares in contrast to 25 hectares for the land cover data set. The EEA is only interested in real changes and not in so called technical changes (changes that are produced by the generalization). Resulting from our experiments in Section 5.3, we plan to intersect versions of the high resolution data DLM-DE and then to filter and aggregate the detected changes.

References

Arnold, S., 2009. Digital Landscape Model DLM-DE – Deriving Land Cover Information by Integration of Topographic Reference Data with Remote Sensing Data. in: Proceedings of the ISPRS Workshop on High-Resolution Earth Imaging for Geospatial Information, Hannover.

Beckmann, N., Kriegel, H.-P., Schneider, R. and Seeger, B. (1990) The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, ACM SIGMOD Rec. 19, 2 (May 1990), pp. 322-331.

Bossard, M., Feranec, J. & Otahel, J., 2000. EEA CORINE Land Cover Technical Guide – Addendum 2000. – Technical Report No. 40, Kopenhagen.

Büttner, G., Feranec, G. & Jaffrain, G., 2006. EEA CORINE Land Cover Nomenclature Illustrated Guide – Addendum 2006. – European Environment Agency.

Douglas, D. & Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10 (1973) pp. 112-122.

Geoff, B. et al. 2007. UK Land Cover Map Production Through the Generalisation of OS MasterMap®. The Cartographic Journal, 44 (3). pp. 276-283.

Gomory, R., 1958. Outline of an algorithm for integer solutions to linear programms. in: Bulletin of the American Mathematical Society, 64(5), pp. 274-278.

Haunert, J.-H., 2008. Aggregation in Map Generalization by Combinatorial Optimization, Vol. Heft 626 of Reihe C, Deutsche Geodätische Kommission, München.

Haunert, J.-H. & Sester, M., 2008. Area collapse and road centerlines based on straight skeletons. in: GeoInformatica, vol. 12, no. 2, pp. 169-191, 2008.

Heinzle, F. & Anders, K.-H., 2007. Characterising Space via Pattern Recognition Techniques: Identifying Patterns in Road Networks, in: W. Mackaness, A. Ruas & L.T. Sarjakoski, eds, Generalization of geographic information: cartographic modelling and applications, Elsevier, Oxford, pp. 233-253.

Jansen, L.J.M. Jansen, G. Groom, G. Carraic, 2008: Land-cover harmonisation and semantic similarity: some methodological issues. Journal of Land Use Science, Vol. 3, No. 2–3, June–September 2008, 131–160.

Kavouras, M. & M. Kokla, 2008. Semantic Integration of Heterogeneous Geospatial Information. In: Advances in Photogrammetry, Remote Sensing and Spatial Information Sciences 2008 ISPRS Congress Book, Li, Z., Chen, J., and Baltsavias, E. (Eds.), CRC Press/Balkema, London, UK, 2008.

Kirkpatrick, S., Gelatt, C. D. Jr., & Vecchi, M. P., 1983. Optimization by Simulated Annealing. in: Science 220 (4598), 671. 13 May 1983.

Kuhn, W., 2003. Semantic Reference Systems. International Journal of Geographical Information Science, Guest Editorial, 17(5): 405-409.

Lillesand, T. M. & Kiefer, R. W., 1999. Remote Sensing and Image Interpretation, 4th edn, John Wiley &\& Sons.

Liu, Y.L., Jiao, L.M., Liu, Y.F.: Land Use Data Generalization Indices Based on Scale and Landscape Pattern. in: Proceedings of the Joint International Conference on Theory, Data Handling and Modelling in GeoSpatial Information Science, vol. 38, no. 2, 2010

Mackaness, W., A., Ruas, A. & Sarjakoski, L.T., 2007. Generalisation of Geographic Information - Cartographic Modelling and Applications, Elsevier Applied Science.

Meng, L., Huang, C., Zhao C. and Lin, Z. (2007) An Improved Hilbert Curve for Parallel Spatial Data Partitioning. Geo-spatial Information Science 10(4), 2007, pp. 282-286.

Pondrenk. M, 2002. Aufbau des DLM50 aus dem Basis-DLM und Ableitung der DTK50 – Lösungsansatz in Niedersachsen. in: Kartographische Schriften, Band 6, Kartographie als Baustein moderner Kommunikation, pp.126-130, Bonn.

van Oosterom, P., 1995. The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning. in: J.-C. Müller, J.-P. Lagrange & R. Weibel, eds, GIS and Generalization - Methodology and Practice, Taylor & Francis, pp. 120-132.

Zhou, X., Abel, D.J. and Truffet, D. (1998) Data Partitioning for Parallel Spatial Join Processing, GeoInformatica 2:2, 1998, pp. 175-204.

GI-Information Generation and Dissemination

GEOSS Service Factory: Assisted Publication of Geospatial Content

Laura Díaz[1], Sven Schade[2]

[1]Institute of New Imaging Technologies, University Jaume I, Castellón, Spain
[2]Institute for Environment and Sustainability, European Commission, Joint Research Centre, Ispra, Italy
laura.diaz@uji.es, sven.schade@jrc.ec.europa.eu

Abstract. Information systems undergo a change from isolated solutions to open infrastructures based on Web Services. Geospatial applications have followed this trend for more than ten years to deal with data representing the status of our environment. International organizations and initiatives promote standards for data encodings and service interfaces that allow establishing Geospatial Information Infrastructures (GIIs). These GIIs provide services to address most of the steps in the geospatial user workflow, such as discovery, access, visualization and processing. However, they do not provide services to assist users in the publication of content. The lack of this functionality challenges the implementation and maintenance of GIIs since publication of content remains a complex task turning GIIs into top-down infrastructures without user participation. In this paper, we suggest extending classical GII architectures with a service that assists in content publication. The Abstract Factory design pattern is used to model this service as a scalable component and the OGC WPS has been chosen as the service interface to increase interoperability. We introduce a prototype as a proof of concept to be evaluated in a forest fire information system.

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_21,

1 Introduction

Numerous geospatial content sources have to be managed to address challenges related to environmental monitoring. Global events, such as forest fires, impact different administration levels and it is important to identify risks and to provide early warning systems at these different geographic scales (de Groot et al. 2006). The current trend is to deploy and organize this information in Geospatial Information Infrastructures (GIIs) also known as Spatial Data Infrastructures (SDIs) (Masser 2005). To increase the efficiency and interoperability of GII many regional and global initiatives work in the establishment of open standards and agreements. A GII on European scale has been legally mandated: the Infrastructure for Spatial Information in Europe (INSPIRE). It should provide environmental data related to 34 themes, including transport networks, land cover, and hydrography, INSPIRE provides important parts of the European contribution to a Global Earth Observation System of Systems (GEOSS).

Initiatives, such as the two mentioned above, describe the overall architecture and best practices to design and implement GIIs. Content is managed by means of regulated and standardized service types. This imposes a distinct life cycle of geospatial content in distributed environments which can be described in four steps as illustrated in Figure 1. First, content must be made available to a distributed system, i.e., content must be published in standard services like discovery and access services. Second, users need to discover content which will be finally accessed by using these services (third step). Finally, users process the content and generate new content, which should be integrated and published in the distributed system closing this cycle.

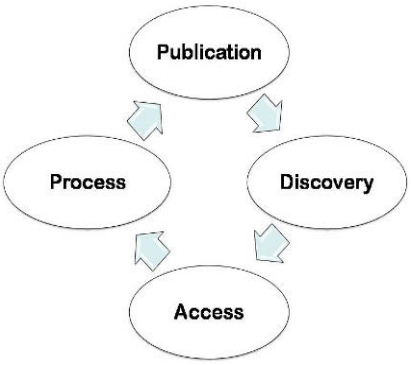


Fig. 1. Content life cycle in GII

From our point of view, GIIs lack mechanisms to assist environmental experts, such as professionals in forest fire modeling, flood prediction, or desertification, and casual users in the publication step. Such mechanisms should assist in making content persistent in a distributed and standard manner rather than storing content locally and isolating it from other users (Díaz et al 2011). Thereby, GII users would become able to publish content in order to maintain an up-to-date GII. Instead, the complex publication mechanisms of traditional GII provoke a low-rate of user motivation regarding participation and content management (Coleman et al 2009). We propose to extend GIIs to address these issues bottom-up. By doing so, we follow one main objective: consideration of user participation to complement the traditional top-down implementations of GIIs by assisting users with publication mechanisms to improve content provision and therefore its availability. Hence, we introduce the GEOSS Service Factory (GSF). GSF assists users in the publication of content in existing standard service instances. It prepares a channel that hides complexity and facilitates content sharing, while remaining loyal to the geospatial initiatives agreements and standards to reach the required level of interoperability. This work extends and generalizes over our previous activities on a GII Service Framework (Díaz et al 2011).

The GSF is our proposal to develop a generic publication service to assist users, both experts of environmental domains or casual users, in content publication on certain systems. To illustrate with a practical example, we show the development and deployment of the GSF in a forestry fires system, where users benefit from this new channel to provide information. Compared to existing Geographic Information System (GIS) tools, which publish geospatial data and maps using proprietary software, the proposed solution is implemented as a standard interfaced web service and can be generally applied a wide range of content types, service standards, and information systems.

The remainder of this paper is structured as follows. Section 2 defines the overall context of geospatial service standards and points to related work. We present the GEOSS Service Factory as part of GII architectures in Section 3, before detailing the involved components (Section 4), the developed prototype (Section 5), and concluding the paper with a discussion and an outlook to future work in Section 6.

2 Background and Related Work

A trend in providing users with the functionality they need is to deploy geospatial applications under service-oriented architecture (SOA) (Papazoglou and Van den Heuvel 2007). One of the goals of SOA is to enable interoperability among existing technologies and provide an interoperable environment based on reusability and standardized components. This approach is focused on an architectural style to design applications based on a collection of best practices, principles, interfaces, and patterns related to the central concept of service (Aalst et al. 2007). The GII paradigm conceptually represents the distributed GIS approach to SOA-based applications in which standardized interfaces are the key to allowing geospatial services to communicate with each other in an interoperable manner, responding to the needs of users (Granell et al. 2010). In this section, we briefly reflect on established global GII initiatives, as well as standard based approaches to geospatial content sharing.

2.1 GII Standards and Initiatives

The purpose of the GEOSS[1] initiative is to achieve comprehensive and coordinated observations of the Earth to improve monitoring and enhance prediction of the behavior of the Earth system. By expressing standard service interface definitions, the GEOSS system assures scalable interoperability. This framework defines GEOSS's common architecture which promotes the use of common principles, techniques, and standards for all GEOSS systems. To improve content and service availability, GEOSS registries allow users to register components and services according to a broad range of standards.

In the European context, the INSPIRE directive sets up a more formalized context for environmental applications and available service types (INSPIRE 2007). It defines a network based on discovery, view, download, transformation, and invocation services. Adopted as a European directive in February 2007, INSPIRE sets out a legal framework for the European GII with regard to policies and activities having environmental impact. INSPIRE is actually based on GIIs which have already been set up and are managed by each member state of the European Commission, thereby creating an infrastructure of GII nodes that are operational at national, sub-national and thematic levels. The technical level provides a range of interoperability standards available for the integration of information sys-

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

tems (Mykkänen and Tuomainen 2008). In our work we will focus on the European context and therefore on INSPIRE. Still, the ultimate goal is to contribute to the broader audience of GEOSS since although it is adopting INSPIRE guidelines, it is more flexible and allows specification of any standard used in a general or particular domain, thereby we consider our approach to be flexible and scalable to be able to publish content in any GEOSS Service.

Within the geospatial domain, interoperability is ensured by standardization efforts most prominently by the Open Geospatial Consortium (OGC). OGC has proposed a number of standards, which promote syntactic interoperability through the use of services (Percival 2008). The existing specifications have been proven to help in setting up GII; these include: OGC Catalogue Services (CS-W), OGC Web Map Service (WMS), OGC Web Feature Service (WFS), OGC Sensor Observation Service (SOS), and OGC Sensor Event Service (SES) interface specifications as examples to mediate geospatial content. Other specifications, such as the OGC Web Processing Service (WPS), provide an interface for accessing processing functionality as distributed Web Services.

INSPIRE specifically recommends the use of OGC Services to implement its service types. An overview of OGC Service Interface Specifications together with superseding INSPIRE Service Types is given in Table 1.

Table 1. Geospatial service types and common standards.

Description	OGC Specification	INSPIRE Service Type
Catalog and Discovery Service	CS-W	Discovery
Portrayal Service	WMS	View
Vector Data Download Service	WFS	Download
Raster Data Download Service	WCS	Download
Sensor Data Download Service	SOS	Download
Web Processing Service	WPS	Invoke
Sensor Event Service	SES	N/A

2.2 INSPIRE-based GII Architecture

The distributed GIS approach to SOA represented by the GII paradigm stimulates the use of standard formats and exchange protocols, and permits the distribution of geospatial functionalities to relevant users (Granell et al. 2010). Moreover, INSPIRE-based architectures offer more benefits to

common technical aspects such as standard interfaces, service types, policies and agreements that enhance data and service interoperability on the European scale.

The INSPIRE technical architecture is a three-layered SOA that differentiates the 'Application' layer, the 'Geospatial Networking Service' layer, and the 'Geospatial Content' layer (Figure 2). The 'Application' layer includes end user applications, ranging from complex Environmental Decision Support Systems (EDSS) to simple clients on mobile devices. Client applications access geospatial content stored in repositories through services in the middle layer. *Geoportals* are a special type of application. They provide the entry point to domain-specific GIIs (Bernard et al. 2005). Such portals are deployed in most of the recent GII implementations.

We already described an initial approach to extend this architecture to increase resource availability (Díaz 2010). A service framework was introduced to integrate bottom-up approaches and top-down methodologies where user generated information and official information could be deployed and published as interoperable services in the INSPIRE-based GII. On the work at hand, we extend this approach by turning the framework into an integrated service component, which exposes its functionality as a standard WPS. Furthermore, we formalize its design and implementation according to the Abstract Factory design pattern (Gamma et al. 1995) in order to increase scalability. More details follow in the next sections.

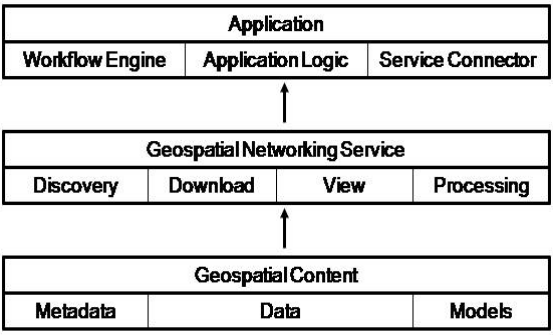


Fig. 2. GII 3-layered Architecture, adopted from (Díaz 2010)

3 Extended GII Architecture

In the same way that GII provide users with functionality to perform daily tasks, such as discovery, access, or download, we propose to provide users

with content publication functionality. This functionality will be provided as an additional web service type deployed in the GII (Figure 3).

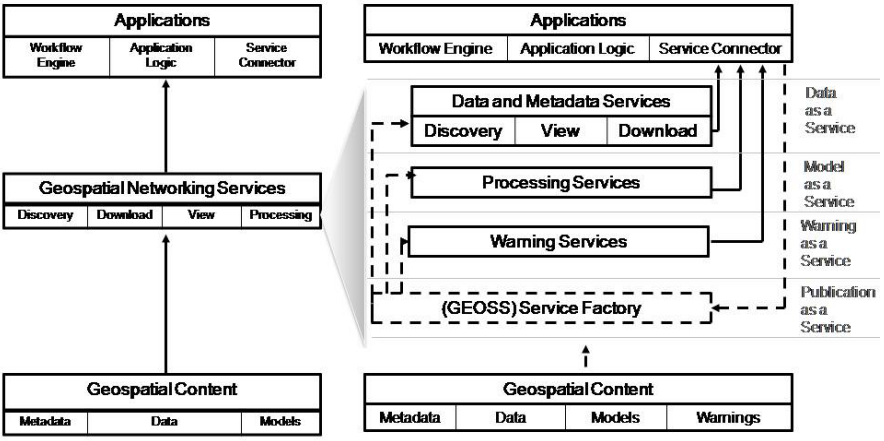


Fig. 3. Proposed extended GII architecture. Left side shows the GII 3-layered SOA, right side shows the proposal to extend the service layer with a new type of service (Publication Service Type)

3.1 GSF Introduction

Following the trend in providing users with data discovery, visualization, downloads, and other functionalities as services, we propose to extend GII with *Publication as a Service* (lower part of Figure 3). The implementing component, called GEOSS Service Factory (GSF) is our proposal to design a standard and scalable publication service. It acts as a mediator to facilitate the publication of new content in GII as standard services compliant with INSPIRE and therefore with GEOSS. The term 'GEOSS' describe its intention to comprehend all the content types and services considered by GEOSS since they are more numerous, diverse, and flexible than INSPIRE. The term 'service' describes its nature, since it is a service and its main function is to update and provide new service content. The term 'factory' defines its behavior; it is designed to be a unique entry point to publish and modify different types of content.

Whereas traditional Geoportals offer the access to the content via the Geospatial Networking Service layer (as seen in Figure 2 and Figure 3), new Geoportals should offer an additional entry point to the GSF as means for content deployment. Scenarios for this advanced publication of content

are manifold; they include the publication of (examples from the forest fire domain are given in parenthesis):

- results of environmental models/simulations (burned area maps);
- model algorithms themselves (burned area analysis);
- warnings created by an EDSS (forest fire warning); and
- geospatial and non-geospatial content from smart mobile devices (geo-tagged photos or simple messages), which we expect to become an important content source in the near future.

All these content types can be integrated in GII using geospatial Web Service technology.

3.2 The Extended Service Layer

On the Geospatial Networking Service layer, a classical GII provides service types for searching, accessing, and processing content that implement the OGC standards, such as WMS, WFS, and WPS. We present those as *Data as a Service* and *Model as a Service* (right part of Figure 3). Recently, those are complemented by warning or alert services, which allow for publish-subscribe mechanisms (IEC/PAS 2004). Although it is currently not included in the INSPIRE development plans, we consider *Warning as a Service*, because push-based message delivery will be required for future GIIs (Schade and Craglia 2010). In a forest fire case, for example, fire brigades and possibly affected citizens should be directly contacted. Such warnings will be deployed by applications in the same way as data content.

As part of the service layer, the GSF mediates between content and the available services. This component is used to assist users and improve the publication stage of the GII content lifecycle defined in section 1. Initially, once the infrastructure is put into place, GSF can be used to integrate first content (dashed arrow on the bottom right of Figure 3). At any later stage, GSF provides the possibility to feed new content from any application (dashed arrow at the right of Figure 3). It publishes the content in the existing services, and optionally updates metadata for discovery (dashed arrows in the middle of Figure 3). In this paper, we aim at sharing geospatial data, environmental models, and warnings. Geospatial data may result from sensor measurements, model execution, or the integration of user-contributed content (Schade and Craglia 2010).

4 GEOSS Service Factory Specification

We now model and formalize the GSF as a mechanism to assist in content publication. This extends an initial approach (Díaz et al. 2010) in two ways, (i) we model the GSF behavior with the Abstract Factory design pattern (Gamma et al. 1995) to make it more scalable and to extend the content types to a broader range and (ii) we wrap this functionality with a standard OGC Service interface. The GSF implements the WPS specification (Schut 2007).

4.1 GSF Design: Abstract Factory Pattern

The *Abstract Factory* design pattern from software engineering (Gamma et al. 1995) is defined as a creational pattern used to instantiate new entities. It encapsulates a group of individual factories that have a common theme. In our case, the GFS holds a group of factories providing operations to publish new entries in Geospatial Services.

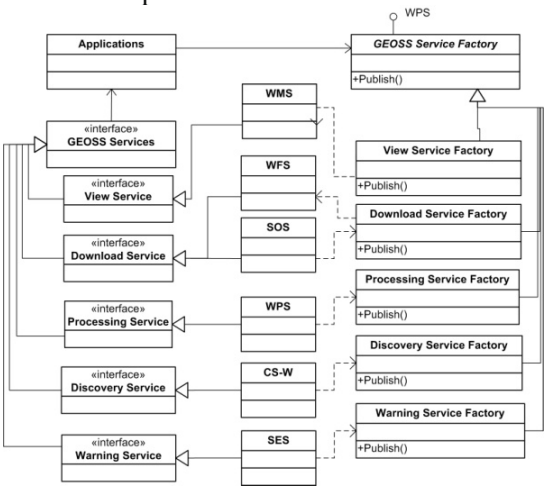


Fig. 4. Abstract Factory pattern applied to (GEOSS) Service Factory.

Figure 4 shows the Abstract Factory diagram adapted to our scenario. We use inheritance to derive the most specific OGC standards from the more general service types defined by INSPIRE and GEOSS (central and left part of Figure 4). View and Download Services are some of the INSPIRE Service Types adopted in our approach to generate GEOSS Services and to deploy them in a certain GII (see also Table 1). The individual services implement a particular OGC standard specification following

INSPIRE guidelines to implement INSPIRE service types. For example, a WMS is used to implement a *View Service* and a WFS is used to implement a *Download Service*.

In this work the scope is limited to publish content exclusively in OGC services; due to their specifications we differentiate two possibilities of communication between the factories and OGC service instances:

1. *Service interface with transactional operations*: Some OGC standards (such as WFS, SOS, and CS-W) include transactional operations in their specification. In this case, the factory can publish content by implementing the client side of the interface. The factory implementation is independent of service implementations. It will be able to deploy content in any instance that implements the chosen (transactional) standard service interface.
2. *Service interface without transactional operations*: Other OGC specifications do not provide transactional operations. In these cases, the publication has to be supported by other means. This is, for example, the case for the WMS, a common standard to implement View Services. The drawback in this case is that the factory depends on the implementation technology of the service instance to publish content.

At this point, it is worth remarking on the special behavior of the *DiscoveryFactory*. This factory can be invoked when the content nature is a metadata, i.e. data about data or services, but it could also be invoked anytime after publishing any other content. We envision the DiscoveryFactory to contain the intelligence to be able to request other service types where new content has just been published in order to extract some metadata and generate automatically a small set of metadata elements to be published in a Discovery Service.

4.2 GSF Interface: WPS for Publication Functionality

The GSF is designed to be implemented as a service component with a standard interface to be re-used in different scenarios. Since the OGC WPS (Schut 2007) is used to reach processing interoperability, it is our standard of choice to implement the GSF as a service. Figure 5 shows the UML class diagram with a simplification of the GSF interface and the signature of the publish process regarding input and output parameters.

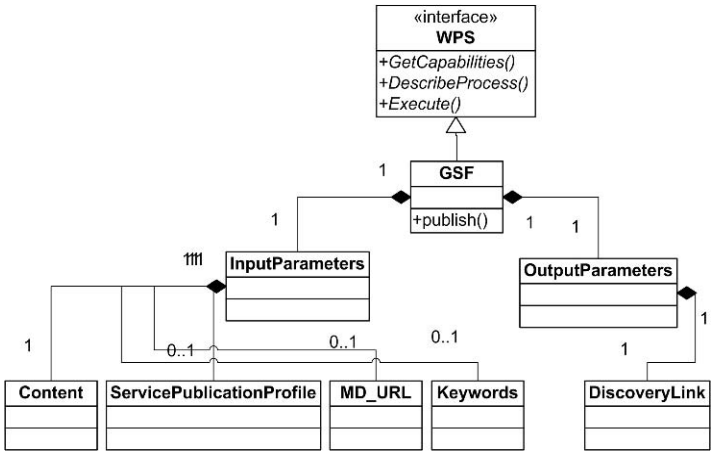


Fig. 5. GSF WPS interface description. This figure shows the input and output parameters of the *Publish* process.

We offer a single process, called *Publish* (see Figure 5). WPS specifies that inputs and outputs can be encoded in many alternative ways. The Publish process considers the following parameters:

- *Content*: Only this input parameter is mandatory. This content can be passed by value or by reference, where a Uniform Resource Locator (URL) to the content can be used. It can vary from a vector or raster data set, a workflow description, or a metadata document.
- *ServicePublicationProfile*: XML encoded parameter that describes the publication policy. This parameter includes information regarding where each data type should be published within this GII.
- *MD_URL*: This parameter indicates that this content is already published in the GII and there are available metadata that should be reused when updating it.
- *Keywords*: The optional 'keywords' parameter provides an initial capability for metadata creation.
- *DiscoveryLink*: This is the only output parameter. This parameter contains the information needed to discover the content published in the system. In the case of the GII, where content is registered in Catalogue services, this parameter contains the end point to the metadata available in the Catalogue Service that contains the description of the content just being published. This contains information about the data services end points serving the content.

4.3 GSF Set-Up: The Service Publication Profile

Each application system deployed in a GII can have its own publication policy. This policy establishes rules, for example, which content type is published in each service type that implement a particular specification and that is located in a concrete end point. At the technical level, we describe this policy as a Service Publication Profile (SPP). SPP is set for a GSF deployed in a GII and it will configure the GSF to decide where each factory publishes the content. For example, the SPP determines which content types are published for visualization and download and also publishes their metadata for discovery purposes.

The use of the SPP allows the GSF to be more scalable and flexible. When invoking GSF, the SPP can describe which content is published for different purposes in different service types. For example, a single geospatial data set may be published in an OGC WMS (View Service) and in an OGC WFS (Download Service) at the same time. Otherwise the SPP can determine that the GSF only publishes this content for one service type. We explain the SPP in more detail in the next section, where we introduce a prototype implementation of the GFS in the context of forest fire management.

5 GSF Prototype: Publishing Content on Forestry

Fire information systems exist from the local to global scale, for example CWFIS (Canadian Wildland Fire Information System) in Canada, AFIS (Advanced Fire Information System) in South Africa, USFS GeoMac in the United States, EFFIS (European Forest Fire Information System) at the European level, and FIRMS (Fire Information for Resource Management System) at the global scale. A common challenge for these systems is to provide geospatial data in a quick and reliable way before and during fire emergencies. It has been demonstrated that geospatial support systems implemented with interoperability standards provide easier access to this type of critical information (Friis-Christensen et al. 2006).

All of these systems offer functionality to access forestry and fire resources in a standard basis. In EFFIS, for example, information layers (raw and processed data) are presented into the map viewer through an internal WMS which will be publicly available among other standard services, such as WFS (Giovando et al. 2010).

Among other functionalities, such as data visualization and download, EFFIS incorporates internal processing, such as burned areas calculation and the forecast of fire danger. Scientific users generate and modify con-

tent, but are IT specialists who have to publish and update this content. As mentioned before, this impedes users' publication and participation and provokes a bottleneck in the publication process. Therefore, we propose to extend EFFIS with GSF as a publication service.

Figure 6 shows the existing EFFIS components based on the Forestry Initial Operating Capacity (FIOC) (EuroGEOSS 2010).

Existing components focus on the content being available through standard services for searching and visualization. We extend the system with a publication service as it is shown in the top of figure 6. GSF is deployed in the system as any other service. As the 'Map Server' and the 'Metadata Server' are associated to their content sources, the GSF is associated to its content sources; these are the (OGC) services in which GSF will publish the content. The publication profile of the EFFIS indicates that the GSF will publish content into the Map Server and the Metadata Server (Figure 6).

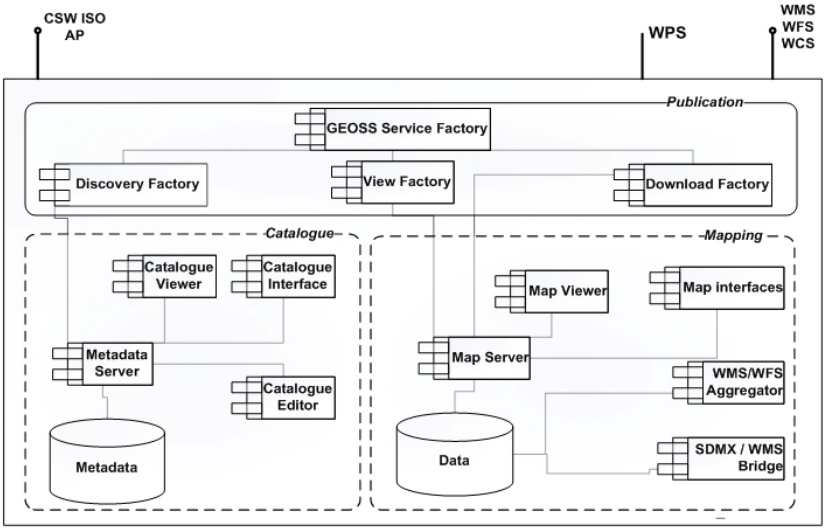


Fig. 6. Example of how to extend an existing system with a Publication Service: GEOSS Service Factory integrated in EFFIS.

At the top layer, GSF has to be integrated into client applications. As we implement the GSF as a WPS, functionality can be accessed through a WPS client. We have implemented such a client as a Java library that is part of the Service Connector (see Figure 2) to be re-used in any Geoportal.

Figure 7 shows an overview of EFFIS extended with the current functionality of the GSF prototype. The figure shows how scientific users can

generate new content, for instance, performing processing (local or distributed), such as the calculation of burned areas, by accessing existing EFFIS services. GSF provides a new channel for assisted publication of certain data types, such as a burned area map. In this way, GSF offers a unique entry point with a standard interface to publish content in the EFFIS system according to a publication policy.

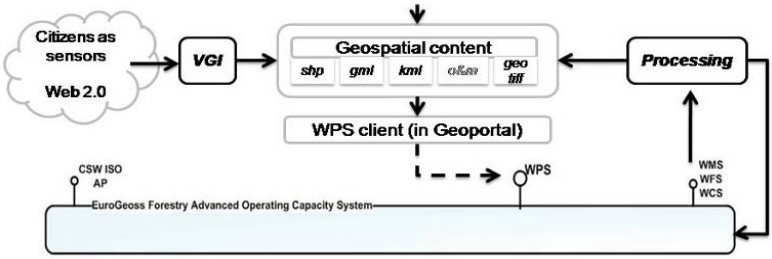


Fig. 7. GSF as a new channel for assisted publication: Different content types tested with current prototype.

5.1 Publishing Forestry Data in GIIs

We have set the Service Publication Profile of the GSF for the forest fires system. View, Download, and Discovery factories are associated with a View, Download, and Discovery service instance deployed in EFFIS. When users generate new content, for instance, by running an environmental model to calculate burned areas, the extended EFFIS provides publication functionality. The system top layer (Geoportal) can connect the user to the GSF (by means of the user interface and the WPS client API as shown in Figure 7). The GSF receives the 'Content' parameter (see Figure 5) and publishes the dataset in a View, Download. and Discovery Service using the individual factories that connect to OGC WMS, WFS, and CS-W instances, respectively. The GSF output is a URL to the newly generated metadata element that is available at the CS-W. This element contains information of how to view and download the dataset published in the OGC services. The procedure is the same for casual users that want to integrate geo-referenced datasets or Volunteered Geographic Information (VGI) in currently supported formats, such as Keyhole Markup Language (KML).

In the GSF prototype we have tested the publication of different kinds of content. We considered the publication of (i) content resulting from internal processing being carried out within the EFFIS system, such as vector data sets with burned area polygons, (ii) NASA images that have to be in-

tegrated in the EFFIS, and (iii) mined data containing geo-referenced news about forest fires coming from social sites and VGI sources.

For the prototype, the *View Service Factory* and *Download Service Factory* publish WMSs and WFSs based on GeoServer technology[2], specifically these factories publish content using the GeoServer RESTful API to connect and update the different service instances. In order to deploy new metadata, the *Discovery Service Factory* implements the transactional interface of the CS-W being independent of the CS-W implementation instance used in EFFIS. For testing purposes we have used Geonetwork[3] as the CS-W implementation.

5.2 Publishing Forestry Models in GIIs

At this stage, we see four possible approaches for publication of environmental models (Schade and Díaz 2010):

(1) Publication of conceptual model descriptions to a repository. For example, the processing steps, required to generate a burned area map, may be described and published as standard encoding of a workflow language. We suggest using the Business Process Modeling Notation (BPMN) (OMG 2009) for this purpose. Such information helps to understand the model to generate environmental data. Discussions on model improvements may be triggered.

(2) Publication of executable files to repository. Following the initial ideas of the Model Web (Geller and Melton 2008) and in line with GEOSS, models may be provided as executables (as *.exe, *.jar, etc). For example, a package for statistical calculations of burned area characteristics can be offered stand alone. Execution will still require download and invocation in a suited environment, but at least models become sharable.

(3) Publication of executable model descriptions to a Processing Service. WPS has been proven as a technology useful to expose and share processing capabilities in Environmental Information Infrastructures (Granell et al. 2010). Among other new functionalities the upcoming WPS 2.0 is considering to publish new processes in running instances. In other words WPS will support the concept of Composition as a Service (CaaS) (Blake et al. 2010).

(4) Publication of existing software to a Processing Service. The option exposes scientific models fully by migrating binary-encoded model

2 http://geoserver.org/display/GEOSDOC/RESTful+Configuration+API
3 http://geonetwork-opensource.org

components as Processing Services (Granell et al. 2010). Distributed computing and in particular the 'mobile code approach' (Fuggeta et al. 1998), in which executable algorithms are sent across a network and executed at distinct nodes, may provide solutions. The relations to grid computing, cloud computing, and virtualization require further exploration.

For the future realization of the Model Web, we assume a combination of all four suggestions getting implemented. As the proposed approaches complement each other, we plan to address them sequentially. Implementations will be guided by the EFFIS example. The GSF will help to increase content availability in GIIs and thereby will aid information discovery and model composition.

5.3 Publishing Fire Related Warnings in GIIs

Other environmental models, such as fire danger forecast and gas emission calculations, are running internally in EFFIS. The results of these models could be used to produce warnings. For this purpose, as an ongoing work, we are experimenting with adding the Warning Service Factory to the GSF. This factory should be able to deploy a new advertisement at an OGC SES (Echterhoff 2008). The *Publish* operator of the SES is used for publishing new warnings to the service. According to the SES functionality, everybody who is subscribed to that type of (warning) event becomes notified. In this manner, we can, for example, create a warning for regions in which the fire danger index exceeds the critical threshold and publish it in a SES instance. Now, affected people (such as the fire brigade and the citizen) can be notified and they can take according actions.

6 Discussion and Conclusions

We argued that content publication to GIIs causes the central bottleneck in environmental information sharing. Bottom-up approaches are needed in the context of GIIs to assist users in populating these infrastructures with content. Following the work of international activities, the GEOSS Service Factory (GSF) was promoted as a solution. We proposed the GSF as a publication service of GII architectures. In order to ensure platform independence, it is provided as a separate component that becomes accessible through standard interface such as OGC WPS from the Geoportal front-end of a GII. In future a possible specialization of this interface can be de-

veloped using profiling to offer a common profile for content publication in GII. In this work, a core operation called 'Publish' and its mandatory and optional parameters have been defined. We recommended the deployment of INSPIRE Service Types, because INSPIRE provides a formal framework to GEOSS while at the same time providing an abstraction layer on top of OGC standards. A prototype has been depicted as proof of concept.

Our proposal alters the role of GII users, being either professionals or casual users. They turn from rather passive consumers into active participants playing a more interactive role and providing new content (Budhathoki et al. 2008). Now, users can participate in the maintenance and updating of the GII. This means that users, besides searching, accessing, and analyzing data, could massively publish newly generated content as interoperable components. This would improve the availability of interoperable content in global, regional and local services related to domain specific scenarios and could increase the effectiveness of GIIs. The monitoring and reporting of forest fires in EFFIS provides one example.

The presented approach raises collateral issues, such as security and quality assurance, which need to be put in place in order to assure the integrity of GII content. This paper already provides a brief outline of how the publication policy in a system could be kept by configuring the GSF with authentication parameters. Furthermore, GSF is an extensible component, which could be extended with modules for content validation.

The suggested flexible implementation of the GSF offers possibilities for resource plug-and-play. New factories can be added using class inheritance. This may be extended in order to publish other content types, such as environmental simulations and VGI, including geospatial data models extended with uncertainty information (Williams et al. 2008) or non-geospatial content, into GIIs. The suggested approach also provides flexibility in terms of functional extensions, such as content validation, security, and automated reasoning capabilities.

First ideas for publishing (environmental) simulations in GIIs have been outlined and we now have to work on the stepwise implementation. We also included existing mechanisms for asynchronous messaging in order to support Warnings as a Service. The OGC SES should be further investigated. Maybe we even have to go beyond the service level to add push-based messaging. Many environmental data is highly dynamic, which requires almost continuous (real time) publishing. As the presented approach publishes content independently of its update frequency, dynamic data also scales in our solution. SOS based data access might be considered.

We did not focus on issues of metadata generation in this paper. As indicated in Díaz (2010), the automated creation of metadata (to be sent to

the geospatial catalogue) is a challenge in its own. First implementations show that it is feasible to extract essential information from some geospatial data encodings, but more detailed elaborations are still required. This area of research defines one of our current activities. We especially investigate the support of metadata documents that follow INSPIRE standards. As the INSPIRE services specifications are more restrictive than those of GEOSS, the global system can be directly fed.

With the presented approach, we move closer to real usage of GIIs, because end users become involved in content provision. Once the barrier of motivation has been overcome, we will be able to benefit from GIIs for effective and efficient information sharing, one of the main goals of GEOSS.

Acknowledgments

This work has been partially supported by the European FP7 Project nr. 226487 called EuroGEOSS.

References

Aalst, W., van der Beisiegel, M., van der Hee, K., Konig, D., Stahl, C. (2007). An SOA-based architecture framework. International Journal of Business Process Integration and Management, vol. 2, issue 2, pp. 91-101.

Bernard, L., Kanellopoulos, I., Annoni, A., Smits, P. (2005). The European geoportal - one step towards the establishment of a European Spatial Data Infrastructure. Computers, Environment and Urban Systems, vol 29, issue 1, pp. 15-31.

Blake, M.. Tan, W., Rosenberg, F. (2010). Composition as a Service. IEEE Internet Computing, vol. 14, no. 1, pp. 78-82.

Budhathoki, N., Bertram, B., Nedovic-Budic, Z. (2008). Reconceptualizing the role of the user of spatial data infrastructure". GeoJournal, vol, 72, pp. 149-160.

Coleman, D., Georgiadou, P., Labonte, J. (2009). Volunteered geographic information: the nature and motivation of producers. In: International journal of spatial data infrastructures research: IJSDIR, vol. 4, pp. 332-358.

Díaz., L. (2010). Improving resource availability for Geospatial Information Infrastructures. PhD Thesis. University Jaume I of Castellón.

Díaz, L., Granell, C., Gould, M., Huerta, J. (2011). Managing user generated information in geospatial cyberinfrastructures. Future Generation Computer Systems, vol. 27, no.3, pp. 304-314

Echterhoff, J. (ed) (2008). OpenGIS Sensor Event Service Interface Specification (SES) version 0.3.0. OGC Discussion Paper, Open Geospatial Consortium.

EuroGEOSS WP3. 2010. D.3.2: Design Specifications for EuroGEOSS Forestry Components and Interfaces. Available at http://www.eurogeoss.eu/wp /wp1/Deliverables/EuroGEOSS_D3-2.pdf

Friis-Christensen, A., Bernard, L., Kanellopoulos, I., Nogueras-Iso, J., Peedell, S., Schade, S., Thorne, C. (2006). Building service oriented applications on top of a Spatial Data Infrastructure: A forest fire assessment example. 9th AGILE International Conference, Visegrad, Hungary, pp. 119–27.

Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns. Addison-Wesley.

Geller. G., Melton, F. (2008). Looking forward: Applying an ecological model web to assess impacts of climate change. Biodiversity 9, no. 3&4.

Giovando, C., Whitmore, C., Camia, A., San Miguel, J., Boca, R., Lucera, J. 2010. Geospatial support to forest fire crisis management at the European level. Proceedings of Gi4DM 2010 Conference on Geomatics for Crisis Management. Torino, Italy, Feb 2010.

Granell, C., Díaz, D., Gould, M. (2010). Service-oriented applications for environmental models: Reusable geospatial services. Environmental Modelling and Software, vol 25, issue 2, pp. 182-198.

IEC/PAS 62030 (2004). Real-Time Publish-Subscribe (RTPS) Wire Protocol Specification, Version 1.0.

INSPIRE EU Directive (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal of the European Union, L 108/1, Volume 50.

Masser, I. (2005). GIS Worlds: Creating Spatial Data Infrastructures. Redlands: ESRI Press.

Mykkänen, J., Tuomainen, M. (2008). An evaluation and selection framework for interoperability standards. Information and Software Technology, vol 50, issue 3, pp. 176-197.

OMG (2009). Business Process Model and Notation (BPMN), Version 1.2. Object Management Group Standard.

Papazoglou, M., Van den Heuvel, W. (2007). Service oriented architectures: approaches, technologies and research issues. The VLDB Journal, vol. 16, issue 3, pp. 389- 415.

Percival, G. (ed) (2008). OGC Reference Model (ORM) version 2.0.0. Open Geospatial Consortium.

Ramamurthy, M. (2006). A new generation of cyberinfrastructure and data services for earth science education and research. Advances in Geosciences, pp. 69-78.

Schade, S., Craglia, M. (2010). A Future Sensor Web for the Environment in Europe. EnviroInfo Conference, Berlin, Germany.

Schade, S., Díaz, L. (2010). Supporting Content Provision in Environmental Information Infrastructures. Environmental Information Systems and Services Infrastructures and Platforms Workshop at EnviroInfo2010, Bonn/Cologne, October 6-8, 2010

Schut, P. (ed) (2007). OGC Web Processing Service (WPS) version 1.0.0. OGC Standard Document, Open Geospatial Consortium.
Williams, M., Cornford, D., Bastin, L., Pebesma, E. (2008) Uncertainty Markup Language (UncertML). OGC Discussion Paper.

Analysis of Quantitative Profiles of GI Education: towards an Analytical Basis for EduMapping

Frans Rip[1], Elias Grinias[2], Dimitris Kotzinos[2,3]

[1]Centre for Geo-Information, Wageningen University, Wageningen, The Netherlands
[2]Department of Geoinformatics and Surveying, TEI of Serres, Serres, Greece
[3]Institute of Computer Science, Foundation for Research and Technology – Hellas, Heraklion, Greece
Frans.Rip@wur.nl, elgrinias@gmail.com, kotzino@teiser.gr

Abstract. There is an ongoing discussion among the members of the GI educational community about the possibility to find a common way to describe a course taught as part of a GI curriculum (anywhere in the world) with the final goal of being able to automatically identify similar courses and define their equivalence. EduMapping is such an initiative that started recently, which used the BoK concepts as its basic labeling scheme. Based on this work we extended the analysis provided by the EduMapping initiative by suggesting and applying an analytical method that is capable of clustering the courses into classes based on (dis)similarity metrics, which are in turn calculated based on the course assessments done by their instructors using the BoK concepts. In this paper, we present and discuss the preliminary results obtained while applying the suggested method on the EduMapping data. We also provide some pointers for further research in an area that has very few contributions so far.

1 Introduction

In 2006 the American Association of Geographers published the Geographic Information Science & Technology Body of Knowledge ('BoK')

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_22,

as a reference for the development of geographic information (GI) courses and curricula (DiBiase, 2006). However, it has not yet been widely accepted as reference, at least not in Europe, as Masik's survey (Masik, 2010) showed. Nevertheless, BoK could provide a basis for a more transparent and more coherent GI education field. The creation of that transparency would require BoK-based description of GI education; creating more coherence would require an understanding of what is common and what is different in today's GI curricula in tertiary education throughout Europe and the world. For that, comparison and analysis are needed, as well as suitable data. Obtaining the data is a problem, because people are not very willing to contribute to surveys. Other obstacles are that information at the academic institutions' web sites is limited and not always available in English or another language known to the researchers; thus the possibility to harvest the necessary data online is also at least problematic. In order to tackle this issue, which is critical for the overall sustainability of our research, we plan to intensify our efforts in data collection by using more international organizations (like AGILE) to get access to their member organizations and provide a comprehensive web site where one could complete the questionnaire and have access to the results in an attempt to more fully exploit the available online information which is improving daily. Finally we plan to better organize the necessary personal communications.

There are two distinct but connected levels of analysis that one can perform on data: analysis about courses and analysis about curricula. The first level is that of the individual courses that need to be categorized according to the BoK concepts (or to concepts that are compatible with it, i.e. BoK extensions and specializations) so that they can be compared with at least some notion of content compatibility. On the second level, curricula should be looked at as a whole and distances between them are calculated. In this way, we try to understand how much one curriculum differs from another based on the courses offered (thus we need to know their compatibility first) and the structure of the educational program as a whole.

The work done in this area so far, as it can be found in the literature, is rather limited. The EduMapping approach has provided a description method with a provisional method of analysis and presentation (Rip and Lammeren, 2010). Its core is a quantitative assessment of the proportion of teaching time spent to BoK-subjects.

In this work it is attempted to provide a unified approach towards analyzing the contents of current GI education using extended datasets from the previous work of Rip and Lammeren (2010). It is realized that the BoK-based approach is basically just a numerical profile of a curriculum or an individual course. A number of such profiles could be collected.

Then the questions are how to compare the profiles and what conclusions to draw. These questions are the subject of this article. The answers to those questions are relevant for further discussion about learning outcomes and competencies, but this is beyond the scope of this particular paper.

In this work we will present results based on data from 11 universities in 7 European countries, thus the conclusions drawn will be rather limited; nevertheless they prove that the suggested methodology has a merit and provides interesting results even on a limited dataset. This creates a basis for testing the methodology in the future with a larger body of data. Finally, the paper is organized in the following way: section 2 presents in more detail the motivation behind this work and refers to work that has already been done; section 3 provides a detailed description of the analytical method proposed and section 4 presents and discusses the experimental results obtained so far. Finally section 5 concludes the work presented here and provides some pointers for further research.

2 Motivation

As described earlier we can focus on two levels in order to better understand what is taught about GI. First we need to tackle the issue at the course level where we want to create a descriptive standard label to add to each GI course description. These labels are a summary of the assessments of the proportion of teaching time spent to BoK Knowledge Areas. Other characterizations like those reported in Painho (2008) or Theodoridou et al. (2008) are also a possibility. Thus, the assigned labels can provide the basis for a content-based classification of the courses taught around Europe. Moreover, they can facilitate the cooperation (discussion and material exchange) among different kinds of GI educators. The wider the acceptance and usage of the labeling scheme, the better our capability to compare GI-courses and GI-curricula, independent from language, country, or organization. This would be in line with the Bologna objective (1999) for the aspects of mobility of students and staff and more transparency for businesses.

There is a second level of information that we deem necessary in order to correctly identify the differences on what is taught among the different departments. We want to be able to compare what is taught in terms of the curriculum as a whole and not just as individual courses. Therefore, we want to be able to measure how far away two curricula are from one another, what is common (and how common), what is different (and how dif ferent) and what is important for a curriculum and whether this is impor-

tant for the other curricula. In the end we would like to be able to understand to a certain extent the differences and similarities among the different curricula and even to be able to identify the existence or non-existence of different "schools" (or clusters) of GI teaching across Europe (or the world), which could give us a better understanding of the influence exercised by different countries pioneering in this area. Another reason for this analysis is the objective to provide more insight in the GI education field for local GI education managers. This level is not part of the work presented in this paper since there were not enough datasets to provide an adequate sampling body and we wanted to actually test the validity of this approach against the EduMapping results presented by Rip and Lammeren (2010).

Ultimately we would be very happy if we could propose an artificial basic curriculum based on the most important subjects identified, although we acknowledge that curricula should be compiled under a strong teacher/student influence and that this work is beyond the scope of this paper. This discussion will be more mature when a more complete profile of both the courses and the categories that the courses are classified into has been created; then we can identify core and elective courses for an artificial curriculum. Thus, in this paper we propose labeling the courses of a curriculum and based on this labeling schemewe extract visualizations and metrics that will help us gain better insights into the (dis)similarity among courses taught throughout Europe.

3 Data Collection and Basic Analytical Methods

3.1 Data Collection

Data collection took place initially via websites and later via personal contacts. Questionnaires were given out where responders were asked for assessments of the proportion of teaching time spent to content subjects mentioned in BoK. The assessments were on individual courses rather than curricula and were done by a person involved in the actual teaching. Here the problems encountered were: (a) not all departments have their curriculum online and not all have an English version or are in a language that we can understand; (b) it was also discovered through the questionnaires that not many of the teaching staff are aware of, let alone familiar with, BoK, so assessments might be superficial; and (c) that sometimes even instructors themselves hesitate to categorize their courses. We collected data for

courses belonging to all kinds of GI education, from commercial 1 day-courses to 3-year academic curricula, and from professional training to Master of Science level. For each curriculum and course we collected the following: name; website; organization or department; ECTS size; name of assessor; proportion of time spent to BoK Knowledge Areas; and time-shares for GI-subjects not in BoK, Generic GI, or non-GI subjects (Rip and Lammeren, 2010). At the time the work reported here started, assessments of 11 curricula and about 40 individual courses from 7 European countries had been collected.

Summary for a curriculum	**ECTS Size**	**Share**
Cat. 1: GI subject groups, mentioned in BoK 2006	**42.5**	**52%**
Knowledge Area AM: Analytical Methods	11.5	10%
Knowledge Area CF: Conceptual Foundations	3.5	3%
Knowledge Area CV: Cartography and Visualization	6.0	5%
Knowledge Area DA: Design Aspects	15.0	13%
Knowledge Area DM: Data Modeling	6.0	5%
Knowledge Area DN: Data Manipulation	1.5	1%
Knowledge Area GC: Geocomputation	1.0	1%
Knowledge Area GD: Geospatial Data	14.0	12%
Knowledge Area GS: GI S&T and Society	1.5	1%
Knowledge Area OI: Org. and Institutional Aspects	2.5	2%
Cat. 2: GI subjects, Not in BoK 2006	**6.5**	**5%**
Cat. 3: Generic	**43.0**	**36%**
Cat. 4: Non-GI subjects	**8.0**	**7%**
total	**120.0**	**100%**

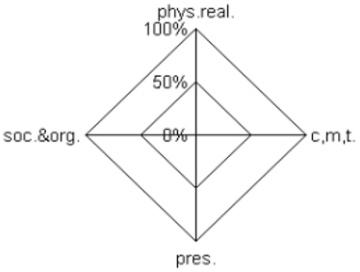


Fig. 1: The table of different "GI-in_BoK" categories and a label with the assessment of a curriculum. Category 1 can be mapped in a 2-dimensional space, with an X-axis going from Society and Organization to Concepts, Methods and Tools. The extremes of the Y-axis are Physical Reality and its Presentation.

3.2 Initial Data Analysis

As defined in Rip and Lammeren (2010), our second objective was to create a map of GI-education in Europe. The map was to be an overview of the pattern of positions of GI-education possibilities as determined by teaching content and not a map showing the geographical location of GI-courses or GI-curricula. The collected data about GI-courses and GI-curricula should therefore be transformed into a two-dimensional position: the map co-ordinates. Rip and Lammeren (2010) realized this spatialization (Skupin and Buttenfield, 1997) by applying two simple analytical steps to the "GI-in-BoK category" (the 10 numbers representing the part of the teaching content that could be expressed in terms of GI-BoK). (see also Figure 1)

The two steps of the analysis were the following

- step 1: conversion of 10 knowledge areas in alphabetic order into a Content Area Polygon (CAP) in a two-dimensional space.
- step 2: calculation of the centroid (centre of gravity) of the CAP as a two dimensional position: the course co-ordinates.

A collection of course co-ordinates (i.e. CAP centroids) was then visualized in a map, showing an overview of course positions in this 2D space (Figure 2).

3.3 Proposed Improvement of Data Analysis

The provisional and rather intuitive nature of this analysis calls for improvement. Due to the course conversion process from a set of ten values per course to one x,y position, there is a considerable loss of detail. This in turn influences the EduMapping results of the mapping process and the position of a course on the final course position map (Figure 2).

In this work we present an analytical approach for charting GIS education, that provides a basis for a more detailed analysis and visualization. This work describes the application of these methods to the data used in the EduMapping approach, with the objective to find out if it could provide an actual improvement for EduMapping. The intention is to produce a grouping of courses in homogeneous and meaningful clusters according to their multivariate description.

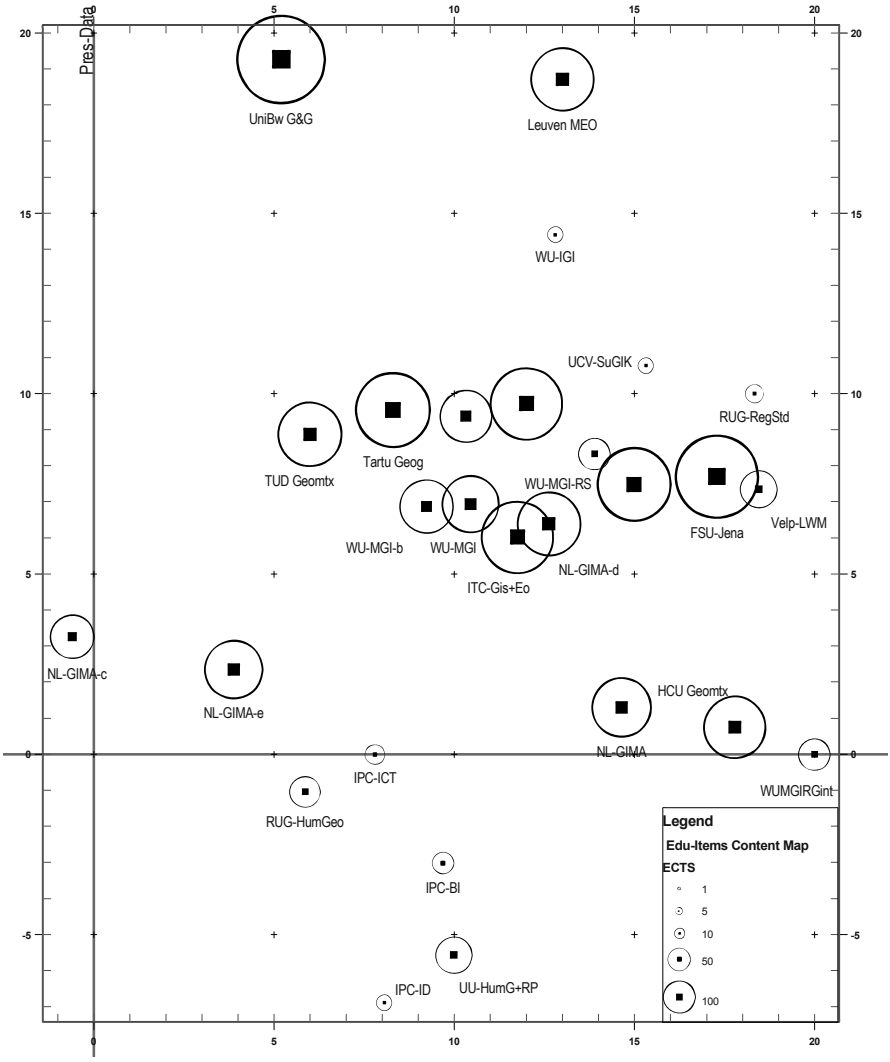


Fig. 2: A map showing course positions (Rip and Lammeren, 2010).

The clustering can be achieved by calculating a quantitative similarity/dissimilarity factor for each course in a curriculum using ECTS and course time-sharing based on GI-in-BoK (viz. Category 1 in Figure 1) as the courses' characterization variables. The known proximity measures that could be used to define this similarity/dissimilarity factor are:

- Dissimilarity Ratio
- Gower's Dissimilarity

- Euclidean Distance (as the crow flies)

If we denote as vectors $\vec{x}$, $\vec{y}$ the (possibly normalized – and in our case it is so) variable values of two courses in a curriculum can be calculated with the formulae below. In these formulae, v is the number of variables or the dimensionality, w_i is the weight for the i-th variable, and $w_i=0$ when either x_i or y_i is missing.

- The Dissimilarity Ratio:

$$d_{DR}(\vec{x},\vec{y}) = 1 - \frac{\sum_{i=1}^{v} w_i x_i y_i}{\sum_{i=1}^{v} w_i x_i y_i + \sum_{i=1}^{v} w_i (x_i - y_i)^2} \quad (1)$$

- The Gower's Dissimilarity:

$$d_{GD}(\vec{x},\vec{y}) = 1 - \frac{\sum_{i=1}^{v} w_i \left(1 - |x_i - y_i|\right)}{\sum_{i=1}^{v} w_i} \quad (2)$$

- The Euclidean Distance.

$$d_E(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{v} (x_i - y_i)^2} \quad (3)$$

Thus, grouping of curricula in homogeneous and meaningful clusters according to their multivariate description is feasible using clustering methods and the distance metrics mentioned above. In the method described in the next section, we actually use Gower's Dissimilarity to measure the distance between courses and classify them in different clusters. Gower's dissimilarity is based on the absolute difference of variables and is thus expected to be more robust than the other two metrics when outliers are present. EduMapping assessments provide ECTS percentages of teaching content elements.

As depicted in Figure 3 for 2 assessments, percentages may be visualized as a horizontal bar of the values of content elements. Each grey value in this plot represents one of the ten BoK Knowledge Areas; the total

width of each bar represents 100% of the teaching time spent to subjects mentioned in BoK, while the width of a grey value represents its timeshare percentage. This way, differences in timeshare among assessments are easily captured.

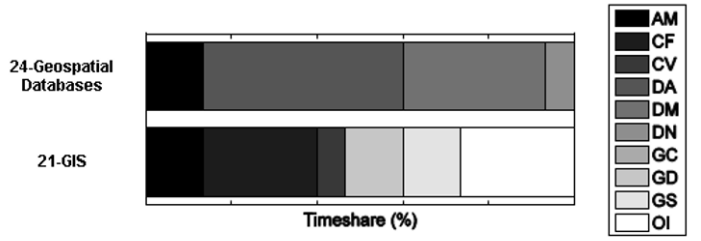


Fig. 3. Horizontal bar plots for assessments 21 (Course "GIS") and 24 (Course "Geospatial Databases").

Grouping of a number M of GI-in-BoK assessments in K meaningful classes is then performed using a clustering technique on the vectors constructed by the 10 Knowledge Area values of each assessment, given the preselected dissimilarity metric between data vectors.

We use a clustering algorithm known as Partition Around Medoids or PAM (Kaufman and Rousseeuw 1990; Theodoridis and Koutroumbas 2006). PAM is considered more robust than the k-means algorithm, because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances. For each class its "central" vector (called medoid) is selected as representative. By contrast, in k-means the statistically computed mean vector of each class is considered the representative vector and this method is more sensitive in the presence of noise and outliers. Both the k-means and PAM algorithms break the dataset into groups. Both algorithms attempt to minimize error in the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm that uses the mean of the cluster, PAM uses points that are centers (medoids) of the cluster and clustering results in medoids. The number of K medoids is selected from the available M course assessments. The mapping of an assessment to a class is based on the minimum dissimilarity of its vector from the medoids. In other words, the courses are clustered around a few of them that are representative for their kind. The results of the clustering are shown in Table 2.

The PAM algorithm works as follows:

Step	Action
1	Read for all assessments the percentages for the Knowledge Areas
2	Initialize: randomly select K of the M data points (assessments) as the medoids
3	Associate each data point with the closest medoid, using the distances as calculated with Gower's Dissimilarity.
4	For each medoid m For each non-medoid data point o Swap m and o and compute the total cost of the configuration
5	Select the configuration with the lowest cost.
6	Repeat steps 3 to 5 until there is no change in the medoid.

The implementation of PAM in the software package LIBRA (Verboven and Hubert 2005) also provides a novel graphical display, the silhouette plot (Figure 4), and a metric called "average silhouette" (Rousseeuw 1987), which is used to dynamically select the optimal number of clusters. This can be done by sequentially 1) performing the algorithm, and 2) computing the "average silhouette" for values of K in an acceptable range, and then picking the result with the maximum "average silhouette" as the best clustering. Thus, the number of clusters is a byproduct of the process and not predefined by the user (the user actually only offers a range of acceptable clusters, i.e. in our case between 2 and 20). After clustering, assessments with a "similar" timeshare pattern are grouped in classes. Differences in the timeshare of content elements between classes of assessments can be easily determined by the bar plot of medoids of the classes.

The Silhouette Plot offers additional qualitative information about the clustering that the PAM algorithm performs, given the set of assessments. Silhouette width indicates the quality of the clustering; a value near -1 in an assessment indicates that the corresponding course is probably placed in the wrong cluster and moreover that one of the adjacent clusters is better suited for that course. A value near zero (0) means that the assessment is placed on the border of two natural clusters, while a value near 1 indicates absolutely correct clustering. We can see, with the notable exception of the third cluster (as we will see this is the "GI Models and Tools" cluster) that three assessments are on the negative side and, all other clusters have either one or no negative assessments, which indicates the quality of the performed clustering. A quantitative expression of this quality is the "Average silhouette", which is the mean of all the silhouette values of the plot.

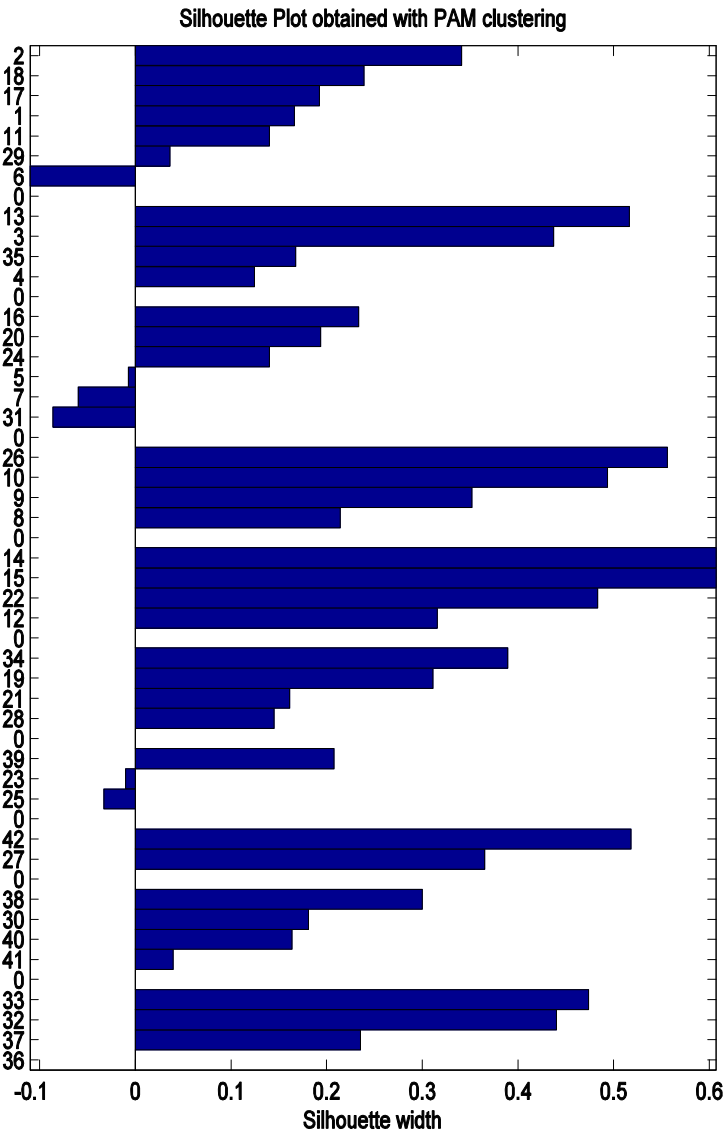


Fig. 4. The Silhouette Plot for our dataset. The 0 ID in the y-axis indicates a class boundary. From top to bottom the clusters 1 to 10 are shown as described in Table 2. Average silhouette of the plot is 0.24.

4 Experimental Results and Discussion

The dataset consisted of 42 assessments for courses completed by people who do the actual teaching of the courses. The timeshare of the 10 BoK Knowledge Areas (as part of the GI-in-BoK category of the course duration) is used to construct the vectors for each assessment and (as discussed) Gower's Dissimilarity is used to measure the distance between vectors. Only the values for the GI-in-BoK category were used and normalized to 100%. Timeshares of the representative assessment for each cluster are graphically depicted by the bar plot of Figure 5. The application of the PAM algorithm has resulted in a proposal to divide the 42 assessments into the 10 clusters listed in Table 1 with provisional names for the clusters.

The created clusters indeed share the same characteristics among the courses that belong to them. For example Cluster #2 named "Cartography" contains courses that range from "Introduction to Cartography" to "Advanced Cartography", which obviously belong together if one would like to create a group of such courses. The courses, though, are not the same. For the algorithm to be able to reach this kind of detailed clustering (so that we can conclude that the contents of a cluster correspond to the same course), we need to have many more courses assessed and included in the EduMapping dataset. Nevertheless we can claim with adequate certainty based on these preliminary results that the algorithm produces promising results and will be able to achieve such detailed classification. We are in the process of collecting more courses and their "GI-in-BoK" assessments.

All the clusters created by the PAM are reasonable. Some courses could, however, belong in more than one cluster and sometimes the names are misleading but we would like to consider the assessment done by the instructors as credible and objective. Having courses belonging to more than one cluster is reasonable and part of the everyday teaching practice since, for example, an introductory GIS course can be classified either under "Introduction to GIS" or "GIS" or even to some extent to "Cartography". The most controversial created clusters are clusters 1 ("Geoinformatics Management and Analysis") and 9 ("Introduction to GIS"). This is expected, since the introductory courses, based on their level of generality and diversity, can easily be classified in different categories but also in the same category. Unfortunately there was no way to assess them as "introductory" since this does not depict their contents.

Table 1. Clustering results from 1 to 10, bottom to top. The numbers in brackets after the course name are the original EduMapping assessment numbers.

Class Label	Class ID	Cardi-nality	Processing ID	Course
Informatics	10	4	33	AppliedInf2
			32	AppliedInf1
			37	Algorithms
			36	Databases
Introduction to GIS	9	4	38	IntroGIS
			30	RS-GIS-Integration (51)
			40	GIS
			41	ThemCartog
GI Modeling	8	2	42	SpDecSuppSys
			27	ModellingInGIS (48)
Geographic Analysis	7	3	39	GeogrAnalysis
			23	GeospatialDataModels (44)
			25	DataMining (46)
SDI	6	4	34	IntroGeoin
			19	SpatialDataInfrastructure (40)
			21	GIS (42)
			28	GIS+LocalCommunities (49)
GIS	5	4	14	IntroRS (35)
			15	IntroGeod+GI (36)
			22	AppsInGIS (43)
			12	AdvGIS (33)
Remote Sensing	4	4	26	RS (47)
			10	GRS20306 (30)
			9	GRS20306 (29)
			8	GRS20306 (28)
GI Models and Tools	3	6	16	GeoinfGIS (37)
			20	GeoinfTools (41)
			24	GeospatialDatabases (45)
			5	ESRI Editor (18)
			7	RS-GIS-Int60312 (48)
			31	SpatModStats (50)
Cartography	2	4	13	AdvCartography (34)
			3	FromDataToMap (3)
			35	IntroCartog

			4	IntroGIS+Carto (5)
Geoinformatics Management and Analysis	1	7	2	SpatialAnalysisWithGIS (3)
			18	GeoinfoManagement (39)
			17	IntroGIS (38)
			1	IntroGeoInformation (1)
			11	GRS20306 (31)
			29	SpatModStats (52)
			6	GRS20306 (25)

One final comment is necessary about the fact that our dataset included assessments for the same course made by different people. This led to an interesting and partially unexpected result: different people teaching the course actually assess it differently and thus, the algorithm classifies it in different clusters. If that happens, the contents of that course may need careful re-assessment or it can be seen as a signal that the description of that course is too open for misinterpretation.

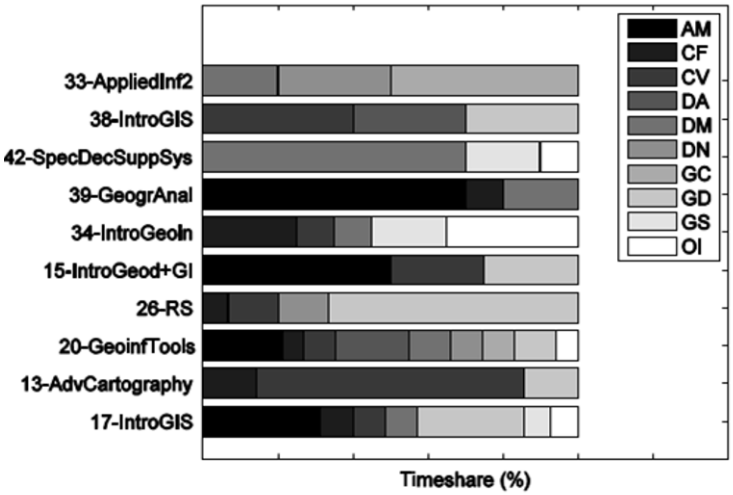


Fig. 5. The representative assessments of all clusters from bottom to top: 1 up to 10.

Figure 5 shows the representative assessments for all clusters. Using this figure we can visually understand the distance among the different clusters. For example, clusters 2 (“Cartography”) and 10 (“Informatics”) have nothing in common and it is impossible to share courses between these two

clusters; the colors (i.e. different BoK Knowledge Areas) are completely different and thus depicting the fact that these two clusters have nothing in common. In terms of BoK, cluster 2 deals with "Conceptual Foundations", "Cartography and Visualization", and "Geospatial Data" (CF-CV-GD). Cluster 10 deals with "Design Aspects", "Data Modeling" and "Geocomputation" (DA-DM-GC) and thus they are completely foreign to one another. On the other hand, clusters 1 and 9 that were introduced earlier share all of the Knowledge Areas present in cluster 9 making them "sister" categories as discussed before. This discussion can be easily extended to the rest of the clusters but since we are only trying to discuss the feasibility and reliability of the method we deem that unnecessary.

In Figure 6 all the assessments of cluster 3 named "GI Models and Tools" are depicted. In this cluster have been placed assessments in which the bigger part of teaching time is devoted to a combination of all Knowledge Areas, except "GS" (GI S&T and Society). We can see that there are assessments that are marginal in the cluster, like assessment 31 (i.e. their 'coloring' probably fits equivalently within another, neighboring cluster). If we look back at Figure 4 ("the Silhouette Plot"), we can see that this information is already shown in that image. This illustrates the importance of the Silhouette Plot for the correct functioning of the algorithm, as well as for understanding its results.

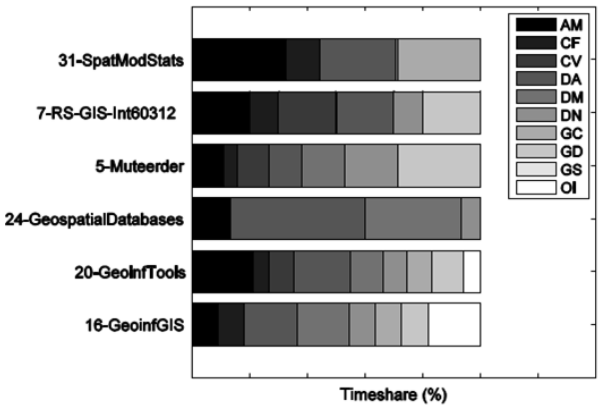


Fig. 6. The timeshares of the 6 Assessments of class 3. Assessment 20 is the representative assessment (medoid) of the class.

Again one could discuss the contents of each cluster and draw interesting conclusions about how we perceive certain issues in GI education but

this is also beyond the scope of this work. Apart from that, the limited quantity of available data does not allow for conclusions without doubts.

One final comment should be made about the dataset itself. It has become obvious from the preceding discussion that the dataset is rather limited, especially if we are trying to draw conclusions about course equivalence. If more data were available we could have "stronger" clusters. We could also have "purer" clusters, namely classes where the contained courses would be equivalent or almost equivalent. The number of clusters might increase. In experiments where even fewer than the available assessments were used, we had even less perfect results – the results improved as the number of assessments (data) that were made available to the algorithm increased. The lack of data makes it difficult to draw final conclusions. Nevertheless, our opinion is that the method provides adequate results even with the limited available dataset.

5 Conclusions

We extended the EduMapping approach by providing an analytical method to analyze the results produced through the mapping process. The method produced a set of clusters of similar courses; the courses, though, cannot be characterized as the same or equivalent due to the limited dataset and the diversity of the courses included. Nevertheless, the method produced clear results in most cases and the courses were placed in the appropriate clusters (it might be somewhat difficult for the reader to figure this out from this paper, since only the course names are indicated. However, going into more detail about the actual contents of the courses in each cluster would be beyond the scope and the size of the paper).

The authors' ambition is to collect, in the immediate future, more data and repeat the analysis for as many curricula as possible, starting with Europe, and trying to involve as many institutions and departments as possible. To validate the method better, we need to collect data from whole curricula so that we can capture differences in courses that depict differences in educational philosophy. With that, it would be possible to draw additional conclusions with regard to the objectives described in the introduction. Examples are to determine the equivalence of courses; determine the importance of courses as indicated by the compactness of clusters; and so on.

The original EduMapping objective to create a map of teaching content positions is not addressed here. Spatialization might be taken up after the

PAM-based methodology proposed here is further developed for full curricula and with a larger body of data.

Finally, more course assessments would help to get closer to discussing what is important in a GI curriculum, described in terms of BoK. It also might be of use for further development of BoK.

References

DiBiase, D., M. deMers, et al., Eds. (2006). Geographic Information Science & Technology Body of Knowledge. Washington, D.C., Association of American Geographers.

Kaufman L., Rousseeuw, P.J. (1990). Finding groups in data. New York, Wiley.

Masik, K. (2010). The usage of UCGIS "Body of Knowledge" in European universities. Presentation AGILE Preconference Workshop, May 11, 2010. Guimaraes, Portugal.

N.N. (1999). The Bologna Declaration of 19 June 1999 – Joint declaration of the European Ministers of Education.

Painho, M. and P. Curvelo (2008). BoK E-Tool Prototype: An ontological-based approach to the exploration of Geographic Information Science & Technology Body of Knowledge. EUGISES. Royal Agricultural College, Cirencester, UK. http://www.eugises.eu/proceedings2008/painho.pdf

Rip, F. I. and R. J. A. van Lammeren (2010). Mapping Geo-Information Education In Europe. ISPRS 2010, Mid-Term Symposium Commission VI - Cross-Border Education for Global Geo-Information, Enschede, the Netherlands. http://www.isprs.org/proceedings/XXXVIII/part6/papers/Rip/Rip+vLammeren.pdf

Rousseeuw PJ (1987). Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20: 53-65.

Skupin, A. and B.P. Buttenfield (1997). Skupin, A. and B.P. Buttenfield. 1997. /Spatial Metaphors for Visualizing Information Spaces. AUTOCARTO 13, ACSM/ASPRS '97 Technical Papers, Seattle, Washington, April 1997, 5, 116-125. http://130.191.118.3/People/Pages/skupin/research/pubs/AutoCarto1997_LoRes.pdf

Theodoridis S, Koutroumbas K (2006). Pattern Recognition 3rd edn, pp 635.

Theodoridou L., Kariotis G., Panagiotopoulos E. and Kotzinos D. (2008), "On Structuring and Sharing Learning Material: A taxonomy for Geoinformatics and Surveying Engineering", Proceedings of EUGISES 2008; Royal Agricultural College, Cirencester, UK

Verboven S, Hubert M (2005). LIBRA: a MATLAB library for robust analysis. Chemometrics and Intelligent Laboratory Systems 75(2): 127-136.

Geotagging Aided by Topic Detection with Wikipedia

Rafael Odon de Alencar[1,2], Clodoveu Augusto Davis Jr[1].

[1]Database Laboratory, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
[2]Serviço Federal de Processamento de Dados (SERPRO), Brazil
odon.rafael@gmail.com, clodoveu@dcc.ufmg.br

Abstract. It is known that geography-aware keyword queries correspond to a significant share of the users' demand on search engines. This paper describes a strategy for tagging documents with place names according to the geographical context of their textual content by using a topic indexing technique that considers Wikipedia articles as a controlled vocabulary. By identifying those topics in the text, we connect documents with the Wikipedia semantic network of articles allowing us to perform operations on Wikipedia's graph and find related places. We present an experimental evaluation on documents tagged as Brazilian states demonstrating the feasibility of our proposal and opening the way to further research geotagging based on semantic networks.

1 Introduction

The expansion of the Internet in recent years has inspired the need for effective information retrieval techniques. Search engines constantly evolve in order to improve their response to user queries taking into consideration various aspects about keywords and their intended meaning. One of those aspects is geography: the user's interest in places is natural, considering that location is a fundamental characteristic of most things, facts, and phenomena. News writers, for instance, strive to clearly provide information as to who, what, where, when, why, and how something happened. Recog-

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_23,

nizing such aspects in documents is a key factor for text-based information retrieval.

The need to obtain or approximate a geographic location for search results is apparent in user queries as they often include place names and other geography-related terms. Previous work shows that queries that include such terms correspond to a significant portion of the users' demand (Delboni et al 2007, Sanderson and Han 2007). Wang et al. (2005) state that many user activities on the Web are directly related to the user's location, so it is important to create applications that take geography into consideration. Much recent work follows this direction with subjects such as the identification of geographic context in Web documents (Alencar et al. 2010), the association of place names to Web pages (Zong et al. 2005), or simply geotagging (Blessing et al. 2007). Successfully accomplishing this task would give us means to enhance current indexing and retrieval mechanisms so that people can search for documents that fall within a delimited geographic scope (i.e. perform local search) (Himmelstein 2005, Schockaert et al. 2008), find nearby services or merchants, or filter content based on regional interests. Service providers would be able to perform geographically-focused advertising and to develop novel ranking strategies for search engines.

This paper shows a technique for tagging text documents with the names of one or more places that describe their geographic scope based on the textual content. Our proposal employs a topic indexing method that uses Wikipedia[1] articles as expected topics (Medelyan et al. 2008). By identifying those topics in a document, it is possible to connect it to Wikipedia, which is used by us as a semantic network. Then we evaluate how the document relates to previously defined target places, information which can also be obtained from Wikipedia, now seen as a source of geography-related content. We demonstrate the feasibility and the potential of the technique through experiments in which news articles are associated to the Brazilian states to which they refer. In a previous work (Alencar et al. 2010), we presented a geographic classification technique based on machine learning, but it showed to be non-scalable for a large number of classes. This paper presents an evolution of the technique towards multiple tagging and introduces the concept of acceptance: we can avoid tagging documents in which there is insufficient evidence of their relationship to any place.

This paper is organized as follows. Section 2 shows related work. Section 3 presents the proposed technique, while Section 4 presents

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

experiments and their results. Section 5 presents conclusions and describes future work.

2 Related Work

Wang et al. (2005) consider three different types of evidence to determine the geographic location(s) associated with a document. First, it is possible to obtain an approximate location of the Web server as informed by services that relate an IP address to a pair of coordinates, such as GeoIP[2]. Second, the location can be inferred by looking at concentrations of users that access the document and their IP-determined locations or by the location of documents that refer to it. Third, the location can be deduced from the textual content of the document. We are particularly interested in the latter, since the location of the Web server can be completely unrelated to the subject of the document and since IP locating techniques are sometimes error-prone and imprecise.

Borges et al. (2007) show that there can be many indications of geographic location in Web documents, but not all of them include unambiguous and easily recognizable evidence, such as postal codes or telephone area codes. Some other works have also focused on identifying the geographic context of Web pages by obtaining references to place names or data such as postal addresses, postal codes, or telephone numbers (Ahlers and Boll, 2008, Blessing et al. 2005), then performing some sort of geocoding (Davis Jr. and Fonseca 2007).

Silva et al. (2006) propose the identification of the geographic scope of a document using machine learning techniques, but warn that doing so directly is hard due to the large number of classes (i.e., locations) and the relatively small number of features that can be used in the classification process. Therefore, they propose a technique that first recognizes geographic evidence in text, and then use a graph-based approach akin to PageRank (Brin and Page 1998) to associate scopes to documents based on a geographic knowledge repository. Such a knowledge base is essential for the process, since it contains information such as place names, postal codes, and even historical names as provided by TGN, the Getty Thesaurus of Geographic Names.

Beyond the recognition of place names, we observe that many other terms can be related to places as well. For instance, terms associated to historical events, monuments, commercial activities, names of authorities,

[2] http://www.maxmind.com/app/ip-location

sports teams, and others can provide indications of geographic location, as long as the semantic connection between the term and the place can be established. Such terms can be used either to establish a location directly or to disambiguate between places that share the same name. The feasibility of this idea was explored by Backstrom et al. (2008) who present a model to track spatial variation in search queries, showing the geographic concentration of the origin of queries related to sports teams. Cardoso et al. (2008) call these terms *implicit geographic evidence*.

Recent work has shown that Wikipedia can be a valuable source of information, considering the semi-structured annotations that exist in its entries and the usual richness of outgoing links, which compose a *de facto* semantic network. Kasneci et al. (2008) present an approach to develop and maintain a knowledge base for which sources are Wikipedia's infoboxes (sections containing attribute and value pairs) and categorical lists, enriched and automatically interpreted with the aid of WordNet[3]. Cardoso et al. (2008) also use Wikipedia to experiment with named entity recognition, and present a system that can find implicit geographic references. Buscaldi et al. (2006) use Wikipedia as a source of geographic evidence and propose building an ontology from geopolitical entities found in the encyclopedia's entries. Buscaldi and Rosso (2007) compare different methods to automatically identify geographic articles in Wikipedia. Wu and Weld (2007) propose an automatic way to "semantify" Wikipedia by extracting its useful data structures. They reveal that, in spite of the incompleteness of the wiki data, systems based on it can be as precise as humans.

Medelyan et al. (2008) describe a method to identify topics in text using Wikipedia articles as a controlled vocabulary. Milne and Witten (2008) present a text enrichment strategy that automatically adds hyperlinks to Wikipedia entries in a process called *automatic wikification* (Mihalcea and Csomai 2007), which matches terms from the text to Wikipedia entries with no natural language analysis.

Alencar et al. (2010) describe strategies for text classification into geography-related categories using evidence extracted from Wikipedia. Terms from article titles and the connection between entries in Wikipedia's graph are used to generate a list of place-related articles, which are then used as features for automatic text classification. Experiments using a news dataset classified over a small subset of the Brazilian states demonstrate that Wikipedia contains valid evidence to be used in the geographic analysis of texts.

In this work, we move one step further from Alencar et al (2010). Instead of matching Wikipedia articles titles in text, we use a topic

[3] http://wordnet.princeton.edu

indexing algorithm (Medelyan et al. 2008, Milne and Witten 2008) that produces a list of articles that best describe the subjects found in the text. We also do not use machine learning for selecting tags, since in previous work we observed a loss of precision when we added more classes, a common issue related to the testing and training steps of such techniques. To avoid that, we explore Wikipedia's graph and propose strategies for selecting place names as tags.

3 Geotagging

This section describes our methodology to geotag documents aided by topic indexing with Wikipedia. We define four tasks presented in detail in the next subsections.

3.1 Defining target places

Wikipedia stands out as a successful example of crowdsourcing: a network of a billion users succeeded in collaboratively creating something hugely useful. By allowing visitors to create, edit, and revise any article, the digital open encyclopedia has become a very popular source of references. Even though Wikipedia's mission statement ("*gathering all of the world's knowledge*") sounds pretentious, the website's success relies on the fact that the available content is being constantly scrutinized by millions of altruistic users. According to Wikimedia Statistics[4], on May 31, 2010, the English version was the largest one with more than 3 million articles, more than 7 million views per hour, and an average of 23 revisions per article per month.

Wikipedia users enrich content by adding interesting features such as *categories* and *infoboxes*. While categories allow us to navigate over the encyclopedia's taxonomy by organizing the articles in common branches of knowledge, code templates help to enrich the text with semi-structured data in specific formats that can be parsed by algorithms. An infobox is a useful template applied over articles linked to the same category consisting of an attribute-value table that is shown to the right of the text (See Figure 1). Many articles include infoboxes, therefore they constitute a powerful resource for automatically mining information from Wikipedia.

Focusing on the location-based perspective of Wikipedia, one can assume that nowadays it contains a large set of place-related articles, many

[4] Wikistats: Wikimedia Statistics - http://stats.wikimedia.org

of which include internal links, infoboxes, and categories. As an example, in the Portuguese version of Wikipedia[5], there are some community projects that focus on editing all articles about Brazilian cities, listing them in hierarchies and in categories and adding infoboxes with useful geography aspects, such as historical and alternative names, adjacent cities, the current mayor's name, some demographic indicators, and even geographic coordinates.

Considering the rich geographic content offered by recent releases of Wikipedia, we can conceive some strategies to define the list of target places to be used in the geotagging process. In the case of Brazilian places, it was possible to manually navigate through the categories system looking for those which list only geographic entities, such as Brazilian states, cities in a state, micro-regions from a state, and even more detailed categories such as city neighborhoods and boroughs. Considering this hierarchical organization, we are able to write scripts to gather all entities listed from each one of these categories, creating a kind of alternative gazetteer in which entities are linked to Wikipedia articles. Infoboxes helped us make sure that an article was really about a certain place, since they provide information such as official names and the name of the parent place in a territorial hierarchy. Observe that this alternative gazetteer will probably miss some places if they are not included in Wikipedia or if they have a low-quality article with no categories or infoboxes. For practical purposes, one can assume that if a place is relevant enough for the web community to be used in geotagging, it will probably have a Wikipedia entry.

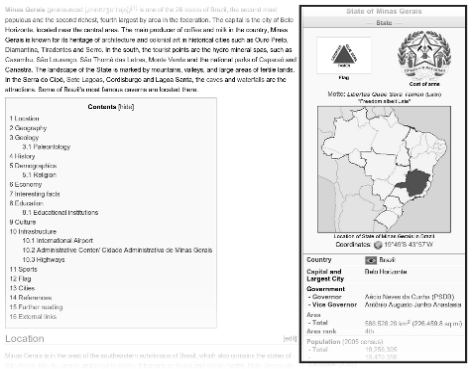


Fig. 1. Wikipedia article about Minas Gerais (Brazilian state), infobox to the right

For our experimental evaluation in this paper, we have considered only Brazilian cities and states. After collecting all such articles from Wikipe-

[5] http://pt.wikipedia.org

dia, we checked them against an official list of Brazilian states and municipalities, and verified that we covered over 99% of the expected number of places.

3.2 Topic Indexing with Wikipedia

Medelyan et al. (2008) described a method to identify topics in text using Wikipedia articles as a source of statistical data about topics and their representative terms. This technique allows us to identify Wikipedia articles as subjects in a document, thus connecting a text with the encyclopedia's semantic network. In the present work, we use this technique to capture the text's relationship to geographic entities found in Wikipedia.

According to the topic indexing method, first all word n-grams from the text must be extracted. Then the probability of an n-gram *a* to be a topic is obtained by counting the number of Wikipedia articles that contain that n-gram as an anchor text (D_{Link}) and dividing by the number of articles that contain that n-gram at all (D_a) (Equation 1). This probability is called *keyphraseness* and its value, ranging between 0.0 and 1.0, can be used to establish a threshold to select n-grams to be included in the topics list.

$$keyphraseness(a) = \frac{count(D_{Link})}{count(D_a)} \tag{1}$$

If an n-gram always points to a single Wikipedia article, then it is chosen to be a topic. Otherwise, if an n-gram has been used to refer to many different articles, we must disambiguate it. This is done by calculating the commonness of a candidate article in relation to a given anchor text and also by calculating the semantic similarity of each candidate entry in relation to other entries that have already been chosen as topics for the text. Commonness is calculated by Equation 2, considering the number of times an anchor text *a* is used to refer to an article *T*, and the number of times it occurs at all. Semantic similarity between two articles *x* and *y* can be calculated by Equation 3, considering their respective hyperlink sets (*X* and *Y*), the overlapping links set ($X \cap Y$), and the total number of articles in Wikipedia (*N*). According to Medelyan et al. (2008), based on Equations 2 and 3, a score can be calculated for each possible article (Equation 4), where *C* is the set of candidate articles for the anchor text *a* in the text *T*.

$$Commonness_{a,T} = \frac{P(a \mid T)}{P(a)} \tag{2}$$

$$Similarity_{x,y} = \frac{\max(\log|X|, \log|Y|) - \log|X \cap Y|}{N - \min(\log|X|, \log|Y|)} \quad (3)$$

$$Score(a,T) = \frac{\sum_{c \in C} Similarity_{T,c}}{|C|} \times Commonness_{a,T} \quad (4)$$

For this disambiguation process, Milne and Witten (2008) go one step further and balance the score formula by applying machine learning to select relevant candidate topics considering the quality of the context along with similarity and commonness.

3.3 Navigating in Wikipedia's graph to reach places

Wikipedia can be seen as a huge graph in which articles represent nodes and the links among them represent directed edges. Since the result of the topic indexing task described in Section 3.2 is the subset of Wikipedia articles that are topics from the text, we can consider our target text document as a new node in this graph.

Our method is based on a breadth-first search on this graph starting from the node that represents the document being geotagged. This kind of graph search begins at a specific node and visits all of its neighbors, then visits all neighbors from the already visited nodes, and keeps going on recursively until it visits the whole graph component. During the visit to a node, we check whether it corresponds to a known place from the predefined list and, if so, we consider it to be a candidate geotag.

When a place is reached in the breadth-first search, we take the current depth of search as the minimum distance from the start node to the place. We then use this information to establish a distance metric (D_p) that quantifies the impact of a certain place p when it is found in the search. If some of the places are already present in the topics of the text document, those will have $D_p = 1$. If a place is found because it is adjacent (in the graph) to some of the original topics in the text, it will have $D_p = 2$.

We use a depth threshold (*maxdepth*) in order to avoid searching the entire graph component, which would naturally lead to much unnecessary and confusing information that is distant from the original context. Notice that when we find a few places in the first depth levels, they will probably contain links to more general places (e.g. states will have links to their corresponding countries). If we keep searching, those general places will also link to other specific nodes, siblings of the ones found first, and that

could decrease our chances to find relevant candidate places. In preliminary tests we observed that a *maxdepth* of more than 2 levels would have covered a large portion of nodes from the almost 1 million nodes of the Portuguese Wikipedia graph. In section 4, we present more details on this issue.

We also propose in our method a metric called *adjacency count* (C_p) of a place node p. During the search, every time we visit a node we need to get a list of all its adjacent nodes in order to feed a queue of nodes to be visited later. If a place node is in this adjacency list, we increase its adjacency count. This metric informs us the relevance of some place to the subgraph obtained at the end of the search. If many topics from the text document are strongly related to a certain place in the Wikipedia graph, then this place's node p will probably get a high value, since many nodes point to it.

The distance and adjacency count metrics for a place p can be combined to generate a final metric called $Score_p$ that decreases the C_p value dividing it by D_p to the power of an *exponential decay* factor (See equation 5).

$$Score_p = \frac{C_p}{(D_p)^{\exp decay}} \tag{5}$$

$Score_p$ represents the relevance of the place in the context of a document, but this relevance gets worse as this place is considered to be farther away from the original topics in the semantic network. As a final result from the search step, we obtain a list of candidate places P, each of which accompanied by a measure of the relevance of the place to the context of the textual document we want to geotag.

3.4 Selecting tags

Given the list of candidate places P, we must now generate a set of tags that define the geographical scope of the text document. First of all, if P is empty, it means that the topics identified in the text could not provide any information on the geographic entities related to the context. In this case the document is not geotagged.

If one or more places are listed in P, $Score_p$ provides us with a *filtering* measure, i.e., using it, we can select the most relevant tags and filter out less relevant responses. We propose some selection strategies with varying selectiveness:

- **Top k:** sort tags downward by $Score_p$, then select the first k tags.

- **Global Threshold:** define a minimum value v for $Score_p$, and return all p in P where $Score_p > v$. $Score_p$ must be normalized in order to define a general-use, percentage-based threshold.
- **Relative to First Threshold:** considering the candidate places list P sorted downward by $Score_p$, calculate the percentage gain g relative to the first place p_1 to the next places p_i using $g_i = Score_{p1} / Score_{pi}$. Define a minimum gain value v and return all p_i so that $g_i > v$. The idea is that the selected tags must be nearly as good as the first one in the ranking. Low g values would allow less important tags to be included in the response, while values close to 1.0 would make it return only those tags in which $Score_p$ is close to the top one. At least the top scorer will be part of the response.
- **Relative to First Threshold with Top-k:** the same as the previous strategy, but limiting the number of responses to the first k ranked.

The process of selecting tags will be the target of future research. For instance, we intend to take into consideration the existence of territory hierarchies, such as country-state-city, so that we can determine the level of the hierarchy to which the text most probably refers.

4 Experimental Evaluation

4.1 Building the Test Collection

For each of the 27 Brazilian states, we performed a manual search for relevant news websites. In such websites, we usually find a section dedicated to local subjects, which we considered to be a good source of texts whose geography scope is limited to a single state. We collected about 100 news articles from each source news site and we read their titles to ensure they were actually about the state. Only the title and body text of the articles were collected. The resulting collection contains about 2700 text documents in Portuguese, each of them labeled with a single Brazilian state (See Table 1).

The topic detection algorithm presented by Medelyan et al. (2008) was adapted to the Portuguese version of Wikipedia using an XML dump of the digital encyclopedia released in March 2010. We have cached the topic indexing output over the collection. Both the raw news texts and the

cached news topics are available at our laboratory's website[6] for free usage.

Table 1. Details about the test collection built with local news from each one of the 27 states from Brazil.

State	Website	Local Section Name	Collection Date	News count
Acre	http://www.agencia.ac.gov.br/	Municípios	July 2010	107
Alagoas	http://www.alagoasnoticias.com.br/	Municípios	July 2010	100
Amapá	http://www.amapadigital.net/	Geral	July 2010	97
Amazonas	http://www.noticiasdaamazonia.com.br/	Cidades	July 2010	100
Bahia	http://www.noticiasdabahia.com.br	Municípios	July 2010	98
Ceará	http://www.cearaagora.com.br/	Cidades, Interior	July 2010	100
Distrito Federal	http://www.correiobraziliense.com.br/	Cidades-DF	July 2010	100
Espírito Santo	http://www.sitebarra.com.br/	Geral	July 2010	100
Goiás	http://www.jornaldaimprensa.com.br/	Estado	July 2010	100
Maranhão	http://www.oimparcialonline.com.br/	Estado	July 2010	100
Mato Grosso	http://www.noticiando.com.br/	Municípios	July 2010	100
Mato Grosso do Sul	http://www.pantanalnews.com.br/	Cidades	July 2010	91
Minas Gerais	http://www.uai.com.br/	Minas	Nov. 2009	104
Pará	http://www.paraonline.inf.br/	Notícias Pará	July 2010	100
Paraíba	http://www.paraiba1.com.br/	Cidades	July 2010	100
Paraná	http://www.parana-online.com.br/	Cidades	July 2010	89
Pernambuco	http://diariodepernambuco.com.br	Vida Urbana	Dec. 2009	104
Piauí	http://piauinoticias.com/	Cidade	July 2010	100
Rio de Janeiro	http://g1.globo.com/rio	Geral	June 2010	100
Rio Grande do Norte	http://www.nominuto.com/	Cidades	July 2010	97
Rio Grande do Sul	http://www.diariodecanoas.com.br/	Cidades/Região	July 2010	100
Rondônia	http://www.rondoniagora.com/	Cidades	July 2010	100
Roraima	http://www.jota7.com/	Roraima	July 2010	100
Santa Catarina	http://www.folhanorte.com.br	All	July 2010	109
São Paulo	http://g1.globo.com/sao-paulo/	Geral	June 2010	102
Sergipe	http://emsergipe.globo.com/	Sergipe	July 2010	100
Tocantins	http://www.anoticia-to.com.br/	Cidades	July 2010	100
Total				2698

4.2 Evaluating Performance

For the first performance test, we applied our geotagging method over the news collection considering only the list of Brazilian states as candidate places. Then we automatically checked if the result tag was the expected state (details in Section 4.2.1). Finally, a manual evaluation using the best parameters from the first test was performed over a small subset of articles, but considering states and cities as resulting tags (Section 4.2.2). Basically, we observe values of *acceptance* (percentage of documents tagged), *macro-f1* (harmonic mean of precision and recall, average of all documents),

[6] http://www.lbd.dcc.ufmg.br/collections

and *accuracy* (number of successfully tagged documents over total tagged documents) on the experiments.

4.2.1 Search Depth Influence on Performance

Our method has a feature that keeps it from tagging a document if no candidate places have been found in the graph search. We evaluated such behavior using the *acceptance* metric, which shows the percentage of the collection that could be tagged. One can imagine that the deeper the graph search, the higher the acceptance rate. However, deeper searches cause the accuracy to drop. In an exploratory analysis, we verified that the Portuguese Wikipedia graph used in our experiments had 1,197,628 nodes with a maximum depth of 61 levels. However, the average distance between any two nodes seems to be small. According to the authors of a tool called *Six degrees of Wikipedia*[7], the average number of clicks to get from any English Wikipedia article to any other is 4.573. Also, by quickly analyzing a cumulative distribution of average nodes reached per level in the breadth-first search, we could see that with a depth of only three levels, more than 50% of all nodes could be reached, describing a typical long tail distribution. Further studies need to be done on this issue. For the present work, we just take into consideration these clues, knowing that there is a trade-off between maximum depth and precision. When the graph exploration goes deeper, it gets harder to select places as geotags.

In order to check the method's behavior with different depth levels, we experimented with the maximum depth set between 1 and 4 with no fine tuning of other parameters and checked the *acceptance* in each one of them. As Table 2 shows, the deeper we get, more documents are accepted by the geotagger. However, the accuracy decreases rapidly; this will be explored in more detail next. Notice that with the maximum depth set to 1, there is no breadth-first search in the graph, only topic indexing. But since the topics have edges between them, this simple adjacency information from the first graph level is already useful to calculate the $Score_p$ of the places, which in this case will be found directly from topics in the texts.

Table 2. Acceptance and accuracy of the method for different max depth levels.

Max Depth	Acceptance	Accuracy
1	60,74%	73,78%
2	98,88%	49,87%
3	100,00%	11,11%
4	100,00%	4,81%

[7] http://www.netsoc.tcd.ie/mu/wiki/

In depth levels greater than 1, we use the exponential decay parameter from Equation 5, so that deeper places have their $Score_p$ value diminished more rapidly according to how far they are from the document in the graph. In Table 3, we considered a maximum depth of 2 and then we explored different values for the exponential decay. We can see that lower values lead to lower accuracy. The highest accuracy that we got was 54.68%, less than the accuracy found with max depth 1. This suggests that geographic evidence found deeper in the graph needs to be treated carefully or it can interfere with the accuracy of the geotagging process. (see also Figure 3)

Table 3. Geotagger performance for different exponential decay values for a *max. depth* of 2.

Exp. Decay	Macro-Precision	Macro-Recall	Macro-f1	Accuracy
0	52.33%	27.07%	27.82%	26.97%
0.5	59.96%	31.60%	33.09%	31.46%
1.0	72.26%	47.72%	49.88%	47.57%
1.5	72.40%	48.83%	50.91%	48.69%
2	70.66%	52.94%	54.07%	52.81%
2.5	71.92%	54.47%	55.76%	54.31%
3	72.01%	54.84%	56.00%	54.68%
3.5	72.01%	54.84%	56.00%	54.68%
4	72.01%	54.84%	56.00%	54.68%
4.5	72.01%	54.84%	56.00%	54.68%
5	72.01%	54.84%	56.00%	54.68%

As we can see, the exponential decay affects the performance in a positive way, since it changes the way $Score_p$ is calculated for places found deeper in the graph search. However, this decay was apparently insufficient to handle both directly and indirectly related places and to achieve better accuracy. The best accuracy found with max depth 2 and varying the exponential decay was 54.68%, against 73.78% using one level, i.e., no depth search. As we said, the number of nodes in the first few levels of depth quickly grows and this can lead to errors. For instance, if an important city is found on the second level and it has numerous links, its adjacency can cause $Score_p$ to overlap the threshold for less important cities found in the first level. To solve this, a more detailed study on how to normalize graph adjacencies needs to be done, so that we can get a positive influence from graph evidence in the geotagging process.

4.2.2 Manual Evaluation

In order to check the geotagger's performance more closely, we evaluated three documents from each state to investigate if the given tags were ade-

quately related to the text. For this evaluation, the set of target places was composed by all states and all cities from Brazil. Instead of the top-1 selection method used in the first experiments, we applied here the Relative to First Threshold, with top-3, and a threshold of 50%. In other words, the geotagger chose the three best tags which scored at least 50% of the first tag's $Score_p$. The geotagger was configured with a maximum depth of 1 and no exponential decay, since this configuration was the best one in the previous tests. Documents rejected by the geotagger were not used, since we wanted to evaluate at least three documents per state.

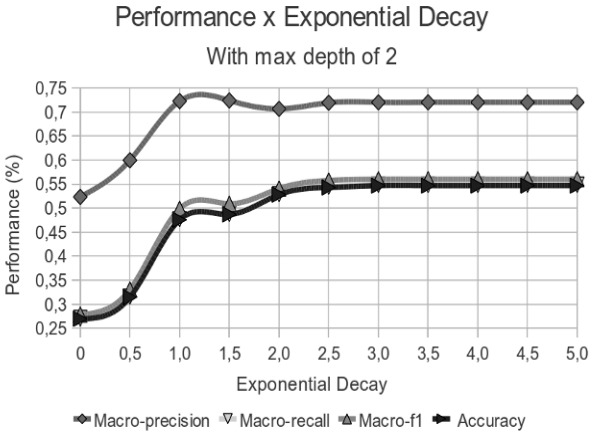


Fig. 2. Geotagger performance for different exponential decay values for a *max. depth* of 2

After applying the geotagging procedure to all documents, we evaluated the tags document by document, reading its title and text, and analyzing the set of places that was returned. If any place was unrelated to the geographical scope of the news, we flagged that document as an error. At the end, we got an accuracy of 78.57%.

We analyzed each error case to consider possible improvements and some interesting cases could be found, especially regarding the Topic Indexing disambiguation process. In one case, the tagger mistook the noun "*campanha*" (Portuguese for campaign) with a city also named *Campanha*. In another case, the city "Rio Branco" from *Acre* state was found as part of a street name, "Avenida Barão do Rio Branco", in another state. Another interesting case regards two states with a similar name: "Mato Grosso" and "Mato Grosso do Sul", causing partial ambiguity and leading to many errors.

5 Conclusions

This paper presented a method for geotagging text based on topic indexing of Wikipedia articles. By identifying Wikipedia articles as topics in the text, we connected the document to the encyclopedia's semantic network and then we used a breadth-first search in the article's graph to obtain a list of candidate places related to the text. Then we applied a scoring technique to determine the best places that should be given as returned tags. During this step, some documents were left untagged due to low geographic information. We think that this rejecting behavior is useful for Information Retrieval mechanisms, since when users ask for documents about a certain location, the search engine should able to retrieve only documents within some estimation of certainty, instead of forcing a classification approach that always associates a document to some place.

In experimental evaluation, we first explored the different levels of depth in the graph breadth-first search step. We noticed that even though deep searches increase the acceptance of documents by providing more geographic evidence, the global accuracy also decreases due to the diversity of places that are to be considered as candidates for tags. We diminished this confusing effect of deep searching by setting up an exponential decay. Nevertheless, the gain obtained by tuning this exponential decay parameter was not enough to make maximum depths greater than 1 useful. Everything indicates that highly connected places in low depths can sometimes mess up the metrics, as their adjacency counts are high enough to overlap less popular but relevant places from the first levels. Thus, our best result was obtained using a max depth of 1 (i.e. only adjacent nodes were considered), getting 73.78% of accuracy and accepting 60.74% of the collection to tag. We believe future work can reach higher gains by carefully exploring other aspects of the Wikipedia graph, such as the relevance and popularity of some entries to balance the given score. In comparison with traditional geotagging approaches, our method can work even in the absence of place names in the text, since we use a semantic network to find topics along the text and gather evidence for geotagging. We intend to perform a more detailed evaluation of this potential in the future.

A more detailed manual evaluation was done with a small set of documents using cities and states as candidates for multiple-tags results. In this case, we could get 78.57% of accuracy and we identified many problems related to the disambiguation process of the topic indexing step. Such errors lead us to consider in future work an experimental fine tuning of the topic indexing method described here and create a biased topic

indexing method focused on high disambiguation precision for place names.

Besides the advantages of using free and up-to-date knowledge created by the Wikipedia community, our technique innovates by making it possible to identify the connection between places and documents, even when they are not explicitly mentioned in the text. Future work also includes experimenting with intra-urban place names, such as neighborhoods, landmarks, and regions as target places.

Acknowledgments

This work was partially supported by the Brazilian National Institute of Science and Technology for the Web (CNPq grant 573871/2008-6), CNPq (grants 474303/2009-8, 551037/2007-5, and 302090/2009-6), Fapemig (CEX-PPM-00168-09), and CAPES, Brazilian agencies in charge of fostering research and development.

References

Ahlers, D. and Boll, S. Retrieving address-based locations from the web. in Proceedings of the 2nd ACM Int. GIR Workshop. 2008. Napa Valley, CA, USA.

Alencar, R. O., Davis Jr, C. A., Gonçalves, M. A. Geographical Classification of Documents Using Evidence from Wikipedia. In Proceedings of the 6th ACM Geographic Information Retrieval (GIR) Workshop. 2010. Zurich, Switzerland.

Backstrom, L., Kleinberg, J., Kumar, R., and Novak, J. Spatial Variation in Search Engine Queries. In International World Wide Web Conference. 2008. Beijing, China.

Blessing, A., Kuntz, R., and Schütze, H. Towards a context model driven German geo-tagging system. in Proceedings of the 4th ACM GIR Workshop. 2007. Lisbon, Portugal.

Borges, K.A.V., Laender, A.H.F., Medeiros, C.B., and Davis Jr., C.A. Discovering Geographic Locations in Web Pages Using Urban Addresses. in Proceedings of the 4th ACM GIR Workshop. 2007. Lisbon, Portugal.

Brin, S. and Page, L. The anatomy of a large hypertextual Web search engine. in Proceedings of the 7th International Conference on the World Wide Web. 1998. Brisbane, Australia.

Buscaldi, D. and Rosso, P. A Comparison of Methods for the Automatic Identification of Locations in Wikipedia. in Proceedings of the 4th ACM GIR Workshop. 2007. Lisbon, Portugal.

Buscaldi, D., Rosso, P., and Peris, P. Inferring Geographical Ontologies from Multiple Resources for Geographical Information Retrieval. In Proceedings of the 3rd ACM GIR Workshop. 2006. Seattle, WA, USA.

Cardoso, N., Silva, M.J., and Santos, D. Handling implicit geographic evidence for geographic information retrieval. in Proceedings of the 17th ACM CIKM. 2008. Napa Valley, CA, USA.

Davis Jr., C.A. and Fonseca, F.T., Assessing the Certainty of Locations Produced by an Address Geocoding System. Geoinformatica, 2007. 11(1): p. 103-129.

Delboni, T.M., Borges, K.A.V., Laender, A.H.F., and Davis Jr., C.A., Semantic Expansion of Geographic Web Queries Based on Natural Language Positioning Expressions. Transactions in GIS, 2007. 11(3): p. 377-397.

Himmelstein, H., Local Search: The Internet is the Yellow Pages. IEEE Computer, 2005. 38(2): p. 26-35.

Kasneci, G., Ramanath, M., Suchanek, F., and Weikum, G., The yago-naga approach to knowledge discovery. SIGMOD Record, 2008. 37(4): p. 41-47.

Mihalcea, R. and Csomai, A. Wikify! : linking documents to encyclopedic knowledge. In Proceedings of the 16th ACM CIKM. 2007. Lisbon, Portugal.

Milne, D. and Witten, I.H. Learning to link with Wikipedia. in Proceedings of the 16th ACM CIKM. 2008. Napa Valley, CA, USA.

Medelyan, O., Witten, I. H. and Milne D. Topic Index with Wikipedia in Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence. Chicago, IL.

Sanderson, M. and Han, Y. Search words and geography. in Proceedings of the 4th ACM GIR Workshop. 2007. Lisbon, Portugal.

Schockaert, S., De Cock, M., and Kerre, E.E., Location approximation for local search services using natural language hints. International Journal of Geographic Information Science, 2008. 22(3): p. 315-336.

Silva, M.J., Martins, B., Chaves, M., Cardoso, N., and Afonso, A.P., Adding Geographic Scopes to Web Resources. Computers, Environment and Urban Syst., 2006. 30: p. 378-399.

Wang, C., Xie, X., Wang, L., Lu, Y., and Ma, W. Detecting Geographic Locations from Web Resources. in Proc. of the 2nd ACM GIR Workshop. 2005.

Wu, F., Weld, D. S. Autonomously semantifying wikipedia. Proceedings of the 16th ACM CIKM. 2007. Lisbon, Portugal.

Zong, W., Wu, D., Sun, A., Lim, E., and Goh, D.H.G. On Assigning Place Names to Geographic Related Web Pages. In Proc. of the 5th ACM/IEEE-CS Joint Conf. on Digital Libraries. 2005. Denver, Colorado, USA.

Specifications for User Generated Spatial Content

Carmen Brando, Bénédicte Bucher, Nathalie Abadie

Université Paris-Est – Institut Géographique National (IGN)-COGIT Laboratory, Saint-Mandé, France
{carmen.brando-escobar, benedicte.bucher, nathalie-f.abadie}@ign.fr

Abstract. This paper addresses the issue of quality in the context of collaborative edition of spatial content. The overall approach is grounded on the definition of explicit and adequate specifications for such content, i.e. the data model, the conceptual model, conventions for data acquisition, possible integrity constraints, possible relationships with external reference data. Explicit specifications could be processed to automatically check when different users simultaneously contribute on the same area. Their definition requires expertness, firstly, to ensure spatial content consistency and, secondly, to establish relevant relationships with external reference data. Designing these specifications is not an easy task for contributors. Hence, the focus of this paper is to assist them in this task. We propose a generic process to automatically produce specification items such as feature types, attribute types, and relationship types, including possible relationship types with external reference data from a set of keywords. It exploits information from two different kinds of existing contents: user generated content (like Wikipedia) and more conventional content (like WordNet and NMA databases). It has been applied to keywords found in existing user generated spatial contents.

1 Introduction

The growth of user generated content (UGC) on the Web has lead to voluminous sources of information like Wikipedia. This trend applies to spa-

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_24,

tial content as well. User Generated Spatial Content (UGSC) stems both from the geotagging of existing UGC, such as Wikipedia articles, and from the edition of geographic features as is done in GeoNames, Wikimapia, or OpenStreetMap (OSM). This kind of content is known within the Geographic Information Science (GISc) community as Volunteered Geographic Information (VGI) (Goodchild 2007). The research community, governmental organizations, and businesses are more and more interested in using UGSC. There may be several motivations to use such data. It is free geographical data, a source for valuable update alerts for mapping organizations, and a source of complementary data which cannot be found in National Mapping Agencies' (NMAs) data sets.

An important stake for usability of UGSC is enhancing its quality. There are several challenges with respect to UGSC: challenges inherited from the "user generated" facet and challenges inherited from the "spatial" aspect. In particular, a major consideration to manage quality of conventional spatial content is to have an explicit structure for the content, e.g. classes and attributes, and conventions that rule unambiguous content acquisition (e.g. the road geometry is acquired at the middle of the road). This information has been designated geographic data set specifications (Abadie 2009). Specifications must be defined carefully to facilitate data consistency. They must be explicit for the user to know how (and how much) the data represent reality. Let us consider someone interested in withdrawing money; for doing so, he/she uses an application to look for ATMs (or cash machines). If he/she does not see ATMs on the map around his location, whereas there are ATMs in nearby areas of the map; he might infer that there is actually no ATM in his neighborhood, and decides to drive to another area in town. Lineage metadata could have been processed by the application to find out that contributors who have edited the map around his location have used a previous version of specifications where ATMs were not already included. In this way, the application could have informed the user that no ATM on this part of the map did not mean there were no ATMs in reality and then, he/she would not have taken the wrong choice. More generally, the relationship between geographical data and the reality cannot be fully assessed without knowing conventions and conceptual models that have ruled data acquisition. Last, user generated spatial content specifications should be formal, at least the semantics part in order to facilitate UGSC integration with other geographic information sources (Kavouras and Kokla 2008). In the remainder of the paper, the term model will refer to conceptual model and data model (ISO 2005).

This research work was carried out within the context of a PhD thesis, which aims at proposing novel methods for improving quality and usability of collaborative geographic data. Our approach is grounded on the im-

provement of specifications (i.e. enhancement and formalization) for this kind of content. Building such specifications is not an easy task for contributors; they possibly do not know enough about GISc to define an adequate model of classes, attributes, integrity constraints, and conventions. Thus, the present paper focuses on our proposal to assist them in this task. We conceive a process to automatically define a set of reusable modeling elements (i.e. feature, attribute, relationship types) dedicated to structuring UGSC. These elements consider relevant clues from two worlds: the world of user generated (spatial) content and the world of conventional geographic databases. The world of UG(S)C consists of large volumes of data available on the Web and of communities of contributors. The relevance of these sources with respect to our concern is that they are crowdsourced. Hence, they make use of non-specialized vocabulary well-known to contributors. The world of conventional geographic databases is typically led by private and public mapping agencies. Relevant hints from this world firstly are their techniques, especially with respect to the modeling of geometry and integrity constraints. Another important hint from conventional geographic databases is data themselves, which have undergone a specific quality checking process and have well documented quality metadata.

The remainder of the paper is organized as follows: section 2 explains briefly the relevance of content structure for quality management in UGSC and then analyzes existing propositions to structure UGC, conventional spatial content, and UGSC; section 3 details the process of building a set of feature types, attributes types, and relationship types for UGSC using several sources of information. It also includes implementation details and some results of this process using an example. Section 4 concludes the paper by recalling its main contributions and by announcing future work.

2 UG(S)C specifications

2.1 Relevance of specifications for UGSC quality management

Several aspects of quality management may be identified in UG(S)C projects (Brando and Bucher 2010; Antoniou et al 2010). Most of them refer to what can be called content specifications.

A first aspect is *internal consistency*. Several components in UGC enhance internal consistency. In wiki-powered sites, the word concerning a relevant concept is an HTTP link to the corresponding page. Whenever

these words are used in a page, the reader can follow the internal link (or wiki link) and obtain the definition of the concept. Some "semantics" may be added to the site by creating categories of articles, which help to reduce possible ambiguities. Internal consistency is also ensured by specific mechanisms to reconcile concurrent editions (Oster et al. 2006). With respect to geographic content, internal consistency also means not having conflicts between geographic features in the database, e.g. a house overlapping a road is usually a topologic conflict. Most of these conflicts are detected by the evaluation of integrity constraints which involve performing spatial operations on data. In other words, management of internal consistency can be enhanced with an adequate structure and explicit integrity constraints, which are part of the content's specifications.

Another aspect is the use of *references to external source*s. Wikipedia contributors are asked to quote external sources. In the world of geographic information, this may designate referencing objects in the real world based on an identifier attribute, for instance. It can also designate a reference to another data set (e.g. a NMA's data set).

A third aspect is a*uthority and reliability of contributors*. In the Google Encyclopedia Knol[1], quality management is mainly based on authors' identification and qualification. In standard metadata for geographic information proposed by ISO, this aspect of geographical data, namely its origin, is described in specific quality metadata: lineage information (ISO 2003). Managing authority and reliability of UGSC contributors has been addressed by Bishr and Kuhn (2007). Abilities of contributors have been empirically analyzed by Budhathoki et al. (2010) in the case of OSM. The authors suggest that those who take part in this open map-making are not laypeople as claimed in recent mainstream GIS literature; most of them have some prior experience in geospatial technology; and they are highly concerned about producing accurate and detailed maps. This seems to suggest contributors are aware of and care about existing specifications, at least in the case of OSM. A fourth aspect is c*omparison with a reference content* whose quality is supposed to be ensured, e.g. comparison between Wikipedia and Encyclopedia Britannica (Gilles 2005). This aspect has been extensively investigated by Haklay (2010) and Girres and Touya (2010). Such a comparison relies on data matching. Having formal specifications should facilitate this matching process because they include formal definitions of the meaning of classes and attributes which can be processed automatically to match data schemas priori to data instances (Kavouras and Kokla 2008; Abadie 2009).

[1] http://knol.google.com

At last, an important aspect of spatial content is the homogeneity of the acquisition process. The space represented by the content must be covered homogeneously. In other words, there should be no loss of balance of geographic space induced by acquisition biases, e.g. in UCrime[2], people are allowed to map criminal activities. Depending on the availability of contributors and their witnessing an assault, an area may be empty of crimes whereas there have actually been more assaults with respect to other neighborhoods. For OSM, Haklay (2010) has also observed heterogeneities in the description of geographic space. Specifications act as unambiguous guidelines for acquisition, hence facilitating the acquisition of homogeneous data. They can also be used to document data, hence to explain heterogeneities related to different versions of specifications used to cover different areas of the map.

This subsection has briefly exposed the relevance of specifications for UGSC quality management. The next subsections present an analysis of existing models to structure UGC, conventional spatial content, and UGSC.

2.2 Models to structure UGC

The best example of UGC is Wikipedia. There are certain elements to structure information within the encyclopedia. An article page explores a single issue and is mainly composed of a title, which summarizes the information concerning the issue in a phrase. It also contains content which discusses the issue in detail.

Furthermore, categories have been defined to annotate articles and organize Wikipedia content. There are many categories related to geography, notably physical geography, which contains sub-categories such as bodies of water, physical infrastructure, landforms, and natural disasters. A category page usually contains a list of the subcategories and articles referencing that category. It may sometimes include a brief description of the category. For example, "the dam category includes articles on dams in general. It includes man-made dams for flood control, hydroelectric power generation, transport, or water supply, as well as natural dams." Articles belonging to the same category may sometimes use a dedicated structure for summarizing information; it has been called an infobox. An infobox is a set of subject-attribute-value triples presenting some common aspects shared by several articles (Wu and Weld 2008). For instance, articles on individual London tube lines include the TfL (or Transport for London)

[2] http://ucrime.com

line infobox, which contains attributes concerning physical characteristics, statistical, and historic information. Similarly to categories, contributors tend to use infoboxes as a way of categorizing articles (Nastase et al. 2010). For instance, many articles concerning the World's mountains use the mountain infobox, which refers to a category.

Specific internal links allow setting up interesting mechanisms for improving the coherence of the entire encyclopedia. Firstly, disambiguation pages are meant to clarify the sense of a certain term (Mihalcea 2007). For instance, the term "plant" possesses several connotations; thus a disambiguation page titled *Plant_(Disambiguation)* has been added. It may refer to “living organisms” or “facility's infrastructures.” Secondly, redirection pages list alternative names for a single issue. For example, body of water and waterbody both have the same signification. Thus, waterbody is actually a redirect page which links to the page body of water. Lastly, Wikipedia is a multilingual encyclopedia which covers more than 25 languages. Every Wikipedia language edition is maintained separately. Pages would usually contain links to the corresponding pages in other languages. For instance, the page for category lakes *(*i.e. *Category:Lakes)* contains a link to the French version *(*i.e. *Catégorie:Lac)*.

An important community effort to extract structured information from Wikipedia is DBpedia[3], which is a knowledge base consisting of over one billion pieces of information from several language editions of Wikipedia. These elements are consistent with a cross-domain ontology, i.e. the DBpedia ontology, which has been manually derived from Wikipedia. DBpedia knowledge base covers general domains of information such as places, persons, organizations, species, etc. However, the coverage of every domain is not exhaustive. It may seem quite superficial for specialized areas of knowledge (e.g. Geography). Another issue is the availability of particular DBpedia data sets in other languages different from English. The most interesting resources (i.e. DBpedia and Infobox ontologies) are only available in that language. They do provide raw data sets in RDF triplet form for infoboxes, article's titles and abstracts, images' description, and internal links in almost any other language.

2.3 Models to structure conventional spatial content

Existing proposals to facilitate the design of geographic conceptual models (ISO 2005; Bédard et al. 2004; Parent et al. 1998) altogether highlight the relevance of feature types, attribute types, relationship types, geometry

3 http://dbpedia.org/

types, level of detail, and temporal aspects. A feature type represents a physical or abstract concept (e.g. road or land lot), and it usually has attribute types defined (e.g. a country's population). A relationship type allows establishing a connection between feature types.

Some of the usual relationships used in GI are composition and specialization relationships. An example of a composition is a feature type individual property which can be composed of a main building and a backyard. Another relevant relationship exclusive to geographic information is the relationship between features that represent the same object but at different levels of details. For instance, it may relate a representation of a city as a point and another representation with a polygon geometry (more detailed). It may also relate this representation of a city as a features collection: a set of buildings that make up the city. Other important relationships in GI are related to topology, distance, and orientation (Bruns and Egenhofer 1996). Most of these relationships are not explicitly specified but can be calculated by performing spatial operations on features' geometries (e.g. containment of districts within cities). Preserving these spatial relationships has always been a matter of concern during evaluation of spatial content consistency. Using a model with shared geometry is a strategy to preserve topological relationships. Another strategy is to use spatial integrity constraints (Mäs 2007). For instance, an integrity constraint indicating that administrative boundary lines are usually placed throughout the middle of waterways can be defined to improve spatial representation of the content.

Besides the definition of a conceptual model, conventional geographic data producers provide a documentation that explicitly describes using natural language how to encode the reality through the conceptual model. They ensure homogeneous capture of content especially when data collectors are different (Abadie 2009). They also help users to understand content (i.e. what to expect from it).

2.4 Models to structure UGSC

Even though part of Wikipedia is spatial content, we distinguish UGSC as content exclusively concerned with the spatial domain. Contributors of UGSC are usually acknowledged as neogeographers in the VGI world, i.e. people who have no academic or professional background in GISc but who are learning through practice (Turner 2006). They play a large role in ordering and categorizing spatial content (Graham 2010). UGSC projects encourage users to use a common vocabulary when editing content, typically by annotating geographic features by means of a user friendly GUI. These annotations are called categories in Wikimapia, tags in OSM, and feature

types in Google Map Maker (GMM). All these annotations will be referred to as tags in the rest of the paper. Examples of tags are historic buildings or state routes. Tags' meaning is documented in the help pages of UGSC projects. For instance, OSM provides all permitted tag values in its wiki pages[4]. Users are encouraged to use already defined tags, though they can freely define their own tags. OSM tags are classified in physical tags for material features, such as highways and waterways, non physical tags for abstract features, such as routes and boundaries, and naming tags for identifying features, such as common and official names of places. GMM distinguishes four main themes, natural features, roads, cities – political regions, and points of interest (POIs). Categorization schemes for both projects seem very exhaustive. For instance, not only pedestrian trail and wetland have been defined, but also less common POIs such as research centers and lighthouses. Wikimapia does not define any themes; all categories are at the same level. At least, this is not clearly available in the documentation pages.

2.5 Summary

To summarize, models to structure UGC, UGSC, and conventional spatial content have their advantages and disadvantages. They can all contribute to the creation of an adequate model to structure UGSC and facilitate its quality management.

UG(S)C consists of large volumes of data available on the Web. This content is crowdsourced by communities of contributors. UGC has been organized through two interesting mechanisms, categorization schemes and internal links. These elements help to enhance internal consistency of the entire content. UGSC tags represent an invaluable source of information about UGSC due to its volunteered nature. They represent a non-specialized vocabulary well-known to contributors, and more comprehensible for neogeographers than the usual argot used in NMAs' databases. OSM model is extensible because it allows contributors to define new tags. UGSC is meant to non-expert contributors but a certain expertise is required to understand the contribution process. Documentation of UGSC models is not quite exhaustive considering that UGSC tags can sometimes be ambiguous. For instance, the difference between mini-roundabouts and roundabouts may be difficult to establish for contributors.

A major consideration for conventional spatial content is to have an explicit structure for such content. It includes feature, attribute, and relation-

[4] http://wiki.openstreetmap.org/wiki/Map_Features

ship types. Examples of relevant relationships and properties for spatial content are composition or topological relationships and the level of details. It also includes conventions and integrity constraints, which rule unambiguous content acquisition. These elements help to enhance homogeneity during acquisition by several operators. Documentation of the content's model plays an important role to solve problems related to ambiguity of certain terms of the model (e.g. for feature types). Another important hint from conventional geographic databases is data themselves, which have undergone a specific quality checking process and their quality is well documented.

3 Proposal: building a predefined model for UGSC

Our proposal aims at facilitating the design of models for UGSC by acquiring predefined modeling elements from diverse sources of information, i.e. feature, attribute, and relationship types. More precisely, we have designed and implemented a process to automatically build modeling elements for UGSC from a set of user keywords. These elements include relationships with external reference data of which quality is known. To illustrate this process, we present an example based on UGSC tags extracted from the main French UGSC projects. The proposed process is also meant to be used on-the-fly to generate new user-defined modeling elements by specifying keywords. This section depicts the process and ends with a discussion about encountered difficulties and some clues to solve them.

3.1 Feature types and attributes types for UGSC

Wikipedia seems a valuable source of information for feature types (categories) and attributes types (infoboxes). Yet the domain of Wikipedia is very wide and not all categories are geography-related. Therefore, UGSC tags (presented in Section 2.4) can be applied as filters to extract relevant Wikipedia categories.

The first step of our approach was to build a filter of geography-related terminology. For our example, we gathered existing UGSC tags from the most popular UGSC projects (OSM, Wikimapia, and GMM). These tags were organized following the GMM theme classification, i.e. natural features, roads, cities and political regions, and POIs. This scheme seems more intuitive than that of OSM. For extracting these tags, the main difficulty was that they can only be manually extracted from help pages. A set

of 432 tags were obtained; there are 66 tags for nature-related elements, 95 for roads and networks, 21 for administrative-related items, and 250 for POIs. This process is illustrated in Figure 1.

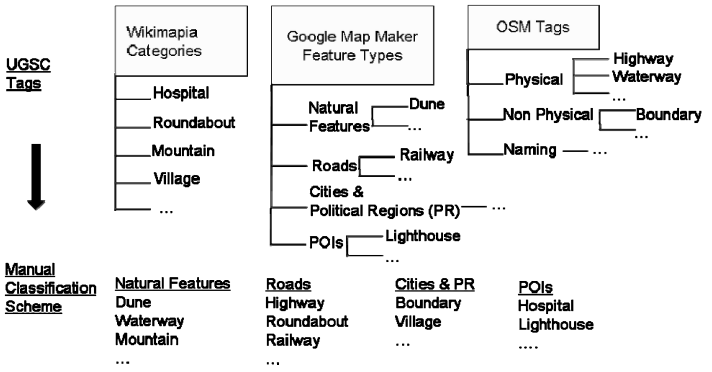


Fig. 1. Step 1: The classification process for an excerpt of UGSC tags (for this paper all tags were translated to the English language)

Importantly, these tags are used to run our process, both to provide initial specification elements and to illustrate the process, but any keyword provided by a user could be used instead of these tags.

The second step consisted in creating features types by querying Wikipedia using the filter described above – or user keywords in the future. We extracted categories and subcategories of these categories. For instance, the category road infrastructure contains the subcategories roads, road bridges, rail trails, road junctions, and pedestrian crossings. Then, for every category, we extracted the corresponding infobox if available. Besides extracting categories, we also retrieved Wikipedia articles for every UGSC tag. They may be considered feature types as well. Indeed, the distinction between categories and articles in Wikipedia should not be systematically interpreted as a distinction between classes and instances (Zirn et al. 2008). For example, a highway would be considered as a feature type, not a geographic feature. Yet, in Wikipedia, there is an article and not a category for highway. Therefore, feature types correspond either to an article or a category in Wikipedia. For our example, the numbers of feature types obtained from Wikipedia using UGSC tags as a filter are presented by theme in Table 1.

The extraction process was carried on by performing string comparison between UGSC tags and titles of Wikipedia pages (and all string comparison of our process), we chose the N-gram similarity measure (Euzenat and Shvaiko 2007), with N=3 considering that most UGSC tags and Wikipedia

pages' titles are usually of small length. Wikipedia data can be manipulated by parsing huge XML database dumps of the entire encyclopedia.

Table 1. Number of feature types created from UGSC tags

	C&PR	NF	Roads	POIs
# UGSC Tags	21	66	95	250
# Feature Types	20	57	37	166

```
IN: theme_j: sets of UGSC tags by theme
OUT: UGSCModel (FTs (ATs, nmaFT, WikiSupCat, WikiSubCat),
                RTs): a UGSC model
Initialize ugscModel ← {}
Initialize currentFT, matchedIB, currentNMAFT ← ''
Initialize currentATs, currentSupCat,
    currentSubCat, currentNMAATs ← {}
for all themes t_i in theme do
  for all tag tag_j in t_i do
    currentFT ← getWikiPage(tag_j)
    currentSupCat ← getWikiSuperCategories(currentFT)
    setWikiSupCat(currentFT, currentSupCat)
    currentSubCat ← getWikiSubCategories(currentFT)
    setWikiSubCat(currentFT, currentSubCat)
    matchedIB ← getWikiInfobox(currentFT)
    currentATs ← getWikiInfoboxAttrList(matchedIB)
    setATs(currentFT, currentATs)
    currentNMAFT ← getNMAFT(currentFT)
    setNMAFT(currentFT, currentNMAFT)
    currentNMAATs ← getNMAATs(currentNMAFT)
    setNMAATs(currentFT, currentNMAATs)
    add(currentFT, ugscModel)
  end for
  for all ft ft_k in ftlist do
    rts ← getHypernymyWordNet(ft_k) +
          getMeronymyWordNet(ft_k)
  end for
  add(rts, ugscModel)
end for
return ugscModel;
```

Fig. 2. Step 2-4: Simplified Algorithm of the process for building a UGSC Model

Considering that we only need information about pages' titles and links between pages, we only queried three relational tables, pages, category, and categorylinks available as SQL dump files (state of October 2010). Querying these tables instead of the XML file solves the difficulties of handling large volumes of content. Nonetheless, the three tables are large in volume as well. Therefore, for optimizing the access to these tables, we created a SQL script which executes delete statements to erase tuples contained in administrative namespaces (e.g. projects, users, etc.). We also tested several indexing structures by measuring processing time and number of disk-block access. These tests showed us that the indexing structures proposed

by Mediawiki provide a reasonable query processing time. This first step of building feature types is summarized in lines 10–15 of a simplified version of the algorithm for the proposed process (Figure 2).

The third step consisted in looking for attributes for our newly created feature types. Wikipedia infoboxes are an important source of attribute-level information. Most infoboxes are retrieved through Wikipedia categories. Yet, there are some infoboxes that are associated only to articles and not to categories. For the human settlement infobox there is both an article and an infobox, but not a category. Next, every attribute specified in the matched infoboxes is assigned to the corresponding feature type. There are two clear issues at this point. First, syntactically similar attributes are repeated in these infoboxes. In this case, a simple merge based on string comparison can help solve it. Second, there are some attributes syntactically different but semantically similar. For instance, state and region are both the primarily administrative division in Germany and France, respectively. Wu and Weld (2009) have built a refined infobox ontology for the English Wikipedia which solves some of these issues. In future work, we plan to include this ontology by automatically translating its concepts using WikiNet (Nastase et al. 2010), which is a multilingual concept network built from Wikipedia.

Infoboxes are incrusted in articles using Wikicode. For instance, the infobox for articles related to rivers is *{{Infobox river| name=val1| ... |river_system=valn}}*, where *val{1...n}* are optionally provided by contributors. Instead of extracting infoboxes' content from the XML Wikipedia dump file, we used the raw infobox data set provided by DBpedia, especially considering that it is the only available information about infoboxes provided in the French language. We retrieved 746 infoboxes from the DBpedia dump (state of March 2010). For our example, we were only able to automatically retrieve 53 relevant infoboxes, leaving more than half of the feature types with no attribute types. This step of building attribute types is summarized in lines 16–17 of the algorithm in Figure 2.

At this step, we also retrieved feature types and attribute types from the model of a specific NMA topographic large scale database. This information was available as a geographic ontology of topographic concepts (Abadie 2009). The detailed description of geometry and attribute types was also available in XML format. In this way, a feature type points to a reference feature type from a NMA model and also contains attribute types retrieved from this reference model. This step is summarized in lines 18–21 of the algorithm in Figure 2. Retrieving these items is interesting with respect to two functions. The first function is to see how an NMA structures a given category of features and to possibly check specific integrity con-

straints. The second function is that the community can use the NMA features as "external references instances" in its model. For instance, the user through the editing GUI could establish a relationship "is within" between a user-defined feature type "restaurant" and an NMA feature type "building." This relationship can be used during edition to check if the restaurants are actually located in buildings.

3.2 Relationship types for UGSC

The fourth step consisted of acquiring relationship types. For this, we have firstly explored the Wikipedia article and category structure. The Natural Language Processing (NLP) community has built two Wikipedia graphs (Zesch and Gurevych 2007): the Wikipedia Category Graph (WCG) and the Wikipedia Article Graph (WAG). The authors provide the following definition: Wikipedia articles form a network of semantically related terms and constitute a direct graph where each node is an article and an edge is an explicit link between articles. Wikipedia categories are organized in a taxonomy-like structure; each category can have an arbitrary number of subcategories where a subcategory is typically established because of a hyponymy or meronymy relationship. However, Hecht and Raubal (2008) explain that most of the relationships in the WCG are limited to hyponymy relations with a sprinkling of meronymy relations.

Therefore, we looked for a more appropriate source to acquire relationship types for clarifying the semantics of the relationship types and for precisely distinguishing composition and specialization relationships. This kind of relationship corresponds to a lexical relationship, i.e. meronymy and hypernymy, respectively. One of the most used resources for discovering lexical relationships between words is WordNet (Miller 1995)[5]. It is a freely available dataset developed for the English language. It has been widely exploited by the research community and integrated in popular dictionary-based websites as a linguistic support (e.g. The Free Dictionary). For the implementation, we used EuroWordNet[6], the European version of WordNet. It has been built by linguistic experts by automatically translating the English version (WordNet 1.5) in a European language like French. This version contains around 22500 synsets shared out between nouns and verbs, including a manual verification. Words or synsets are interlinked by means of conceptual-semantic and lexical relations, such as hypernymy/hyponymy, meronymy/holonymy, synonymy, and antonymy. By

[5] http://wordnet.princeton.edu
[6] http://www.illc.uva.nl/EuroWordNet

exploring this resource, we were able to extract relationship types among the entire set of feature types extracted in the previous step. We were particularly interested in hypernymy and meronymy relationships which play an important role when evaluating spatial integrity constraints on geographic data. This step of building relationship types is summarized in lines 24–28 of the algorithm in Figure 2. An excerpt of the meronymy relationships created for the proposed example is illustrated in graph form, in Figure 3. The resulting directed graph consists of nodes and edges representing feature types and relationship types (i.e. hypernymy/meronymy relationships), respectively. For instance, *sidewalk* → *road* means that roads are composed of sidewalks.

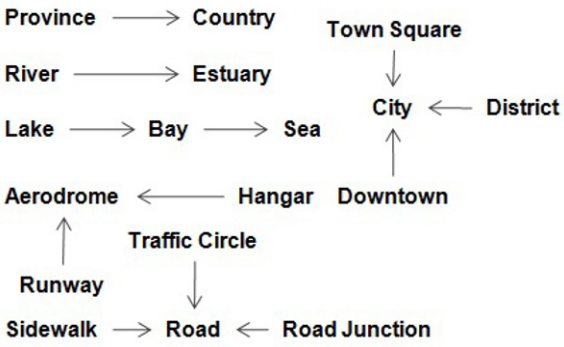


Fig. 3. Excerpt of meronymy relationships created for the proposed example (tags were translated to the English language)

3.3 Discussion

In general, we have found several issues of structure and consistency from Wikipedia, which are being investigated by the NLP community. Therefore, our approach inherits some of these limitations. Notably, this community needs to tackle particularities of languages different than English.

The proposed approach is also limited by WordNet coverage of relationships. This may be solved by adding another expert-validated source of information which can provide new relationship types to the UGSC model. A solution may be provided by Cyc[7], which is a large scale knowledge repository of everyday common sense knowledge. Concerning spatial content, an issue in WordNet is the relatively small amount of meronymy relationships with respect to hypernymy ones. That is, there is a large number

7 http://www.cyc.com

of specialization relationships and a small amount of composition relationships. It is unfortunate since the latter are of high importance for enhancing consistency of the content when evaluating spatial integrity constraints.

Another issue is the relatively small amount of Wikipedia infoboxes. That is, there is no guaranty that all matched categories or articles will have infoboxes. For improving infobox templates, the French Wikipedia has created the project Infobox Version 2[8]. They expect to enhance the definition of infoboxes, increase their coverage, and merge redundant infoboxes. This project can bring light to our issue of an insufficient number of attribute types. We have also considered to translate to the French language the refined infobox ontology provided by Wu and Weld (2009). This will allow solving the issue of insufficient coverage of Wikipedia infoboxes. The concepts of this ontology will be automatically translated using WikiNet (Nastase et al. 2010), which is a multilingual concept network built from Wikipedia.

4 Conclusion and future work

In this paper, we presented a novel proposition for managing the quality of UGSC based on enhanced specifications. To assist contributors in building such specifications, we have developed a generic process for structuring UGSC by yielding relevant modeling elements from user keywords. These elements are feature types, attributes types, and relationships types. For creating feature types, Wikipedia articles and categories are retrieved by applying a geography-related filter, which is derived from UGSC tags. Afterwards, for every newly created feature type, infoboxes, super- and subcategories are extracted from Wikipedia. Next, attribute types are created for every attribute of the matched infoboxes. For acquiring relationship types, the lexical relationships hypernymy and meronymy are queried in WordNet for every newly created feature types. Feature types, relationship types, and integrity constraints from an NMA data set are also retrieved, that can be used both to get suggestions about how to structure a particular content and to make relationships between UGSC and NMA content. Our proposed process allows users to take the best from two worlds when structuring their content- the world of UG(S)C and the world of conventional geographic databases. The preliminary model obtained from current tags of UGSC projects will be available on demand. These results can then be evaluated or compared to other UGSC proposed specifications. In fu-

[8] http://fr.wikipedia.org/wiki/Projet:Infobox/V2

ture work, we plan to perform user tests to investigate whether the proposed method actually helps users to build a model for their spatial content. Moreover, we will investigate the reconciliation of distributed operations on UGSC.

Acknowledgements

The authors are grateful to the reviewers of the submitted version of this paper for their most valuable inputs and comments to improve it.

References

Abadie N (2009) Formal Specifications to Automatically Identify Heterogeneities, in the 12th AGILE International Conference on Geographic Information Science Pre-Conference Workshop: Challenges in Spatial Data Harmonization, Hannover, Germany

Antoniou V, Haklay M, Morley J (2010) A step towards the improvement of spatial data quality of Web 2.0 geo- applications: the case of OpenStreetMap, in the 18th GISRUK Conference, London, UK, pp. 197–201

Bédard Y, Larrivée S, Proulx MJ, Nadeau M (2004) Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentation on Perceptory. In: Wang S et al. (eds) Conceptual Modeling for Geographic Information Systems Workshop, LNCS 3289, pp. 17–30

Bishr M, Kuhn W (2007) Geospatial Information Bottom-Up: A Matter of Trust and Semantics, in: Fabrikant SI, Wachowicz M (eds) The European Information Society - Leading the Way with Geo-information, Springer Verlag LNGC, pp 365–387

Budhathoki N R, Nedovic-Budic Z, Bruce B (2010). A framework for volunteered geographic information: Proposal and illustration. Geomatica 64 (1): 11–26

Brando C, Bucher B (2010) Quality in User Generated Spatial Content: A Matter of Specifications, in the 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal

Bruns HT, Egenhofer MJ (1996) Similarity of Spatial Scenes, in Proceedings of the 7th International Symposium on Spatial Data Handling, pp 31–42

Euzenat J, Shvaiko P (2007) Ontology Matching, Springer-Verlag, Berlin Heidelberg, p. 73

Gilles J (2005) Internet encyclopedias go head to head. Nature 438(7070): 900–901

Girres JF, Touya G (2010) Quality Assessment of the French OpenStreetMap Dataset. Transactions in GIS 14 (4): 435–459

Goodchild M (2007) Citizens as Sensors: The World of Volunteered Geography. GeoJournal 69(4): 211–221

Graham M (2010) Neogeography and the Palimpsests of Place. TijdsCHrift voor Economische en Sociale Geograpfie 101(4): 422–436

Haklay M (2010) How good is OpenStreetMap information? A comparative study of OpenStreetMap and Ordnance Survey datasets for London and the rest of England. Environment and Planning B 37(4), pp. 682 – 703

Hecht B, Raubal M (2008) GeoSR: Geographically Explore Semantic Relations in World Knowledge, in Bernard L, Friis-Christensen A, Pundt H (eds) 11th AGILE International Conference on Geographic Information Science, Springer Verlag LNGC, pp 95–113

ISO (2003) Geographic Information - Metadata, International Standard, TC211/ ISO19115:2003

ISO (2005) Geographic Information – Rules for application schema, International Standard, TC211/ ISO19109:2005

Kavouras M, Kokla M (2008) Theories of geographic concepts – Ontological Approaches to Semantic Integration. CRC Press

Mäs S (2007) Checking the Integrity of Spatial Semantic Integrity Constraints, Constraint Databases, Geometric Elimination and Geographic Information Systems

Mihalcea R (2007) Using Wikipedia for Automatic Word Sense Disambiguation, in Proceedings of the North American Chapter of the Association for Computational Linguistics, Rochester, USA

Miller GA (1995) WordNet: A Lexical Database for English, Communications of ACM 38(11): 39–41

Nastase V, Strube M, Boerschinger B, Zirn C, Elghafari A (2010) WikiNet: A Very Large Scale Multi-Lingual Concept Network, in Proceedings of the 7th International Conference on Language Resources and Evaluation, Valletta, Malta

Oster G, Urso P, Molli P, Imine A (2006) Data consistency for P2P collaborative editing, in Proceedings of the ACM Conference on Computer-Supported Cooperative Work, pp. 259–267, Alberta, Canada

Turner A (2006) Introduction to Neogeography, O'Reilly Media, p 2

Parent C, Spaccapietra S, Zimanyi E, Donini P, Plazanet C, Vangenot C (1998) Modeling Spatial Data in the MADS Conceptual Model, in Proceedings of the 8th International Symposium on Spatial Data handling, Vancouver, Canada, pp. 138–150

Wu F, Weld DS (2008) Automatically Refining the Wikipedia Infobox Ontology, in Proceedings of the 17th International World Wide Web Conference, Beijing, China, pp. 635–644

Zesch T, Gurevych I (2007) Analysis of the Wikipedia Category Graph for NLP Applications, in Proceedings of the TextGraphs-2 Workshop, pp 1–8

Zirn C, Nastase V, Strube M (2008) Distinguishing between Instances and Classes in the Wikipedia Taxonomy, in Bechhofer S, Hauswirth, Hoffmann J, Koubarakis (eds) The Semantic Web: Research and Applications, 5th European Semantic Web Conference, Springer Verlag LNCS, pp 376–387

An Empirical Study on Relevant Aspects for Sketch Map Alignment

Jia Wang, Christoph Mülligann, Angela Schwering

Institute for Geoinformatics, University of Muenster, Muenster, Germany
{jia.wang, cmuelligann, schwering}@uni-muenster.de

Abstract. Sketch maps are drawn from memories and they are in general schematized and distorted. However, the schematizations and distortions are not random. They are a consequence during the cognitive process of perceiving, memorizing, and producing spatial layout. This paper describes an empirical study to investigate the impact of distortions on similarity perception. The study is designed as a human-subjects experiment of similarity ranking with two scenarios. Subjects were presented with 45 sketch maps and one reference map in each scenario; they were asked to rank the sketch maps according to their similarities with the reference map. The results of the experiment are used to develop a cognitively motivated alignment strategy for computer-based comparison of sketch maps and metric maps.

1 Introduction

The last few years seen a substantial growth in user-generated content that is firmly related to Geographic Information Science (GIS) and collaborative Internet applications such as Google Map, Wikimapia, and OpenStreetMap, which allow users to contribute online geographic information. Such volunteered geographic information (VGI) has profound impacts on GIS, particularly on the local level of various geographic locations that go unnoticed by authoritative mapping agencies. Though VGI applications opened more capabilities of GI systems to the general public, the technical requirement for contributing geographic information

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_25,

voluntarily is still demanding; technologies such as Web 2.0, Georeferencing, Geotags, high-quality graphics and their computing and dynamic visualization, and broadband communication are still required to make VGI possible (Goodchild 2007). An interface using sketch maps to contribute geographic information meets the current requirement of VGI systems. Most people can sketch a map of a local environment to update or query a spatial database; computers, Internet and the knowledge for Geotags and Georeferencing are not necessary. Users can simply take a picture of a sketch map and send it as an MMS to a server. Thus, an intuitive human-computer interface is necessary to ease the way that users interact with VGI applications, as well as lessen requests for technologies from the user side.

Sketching a map is an intuitive way of communication about spatial information. Everyone has associations with the area he/she lives in and creates a mental spatial layout of his/her environment. It provides a framework for wayfinding and navigation or serves as a general view of an area. Over the last several decades, sketch maps have been externalized for studying individuals' environmental knowledge. They reflect human spatial thinking, so they are especially effective in tasks that involve spatial information. Accordingly, sketch maps have been utilized frequently by environmental psychologists and geographers to investigate how humans represent spatial information (e.g., Ladd 1970; Beck and Wood 1976; Montello et al. 2005; Tversky 2003, 2005). Asking subjects to produce a two dimensional view of an area from an aerial perspective was already demonstrated as a reliable data collection method; the same individual will produce essentially the same sketch map of the same area over a short period of time despite varied instructions and familiarity with the area to be sketched (Blades 1990). In the last decade, applications using sketches were developed in the GIS domain, e.g., Blaser and Egenhofer (2000) proposed a visual tool using sketches to update or query a spatial database; Forbus and his colleagues (2003) developed CogSketch, which is a general architecture for sketch understanding built on nuSketch (Forbus et al. 2001). The existing sketch interpretation approach *query-by-sketch* (Egenhofer 1996; Blaser and Egenhofer 2000) and the sketching applications as CogSketch are not based on cognitive insights on human spatial thinking. This paper addresses this gap and investigates an alignment strategy accounting for human spatial cognition. The paper describes a human-subjects experiment to discover the impact of sketch map distortions and schematizations on similarity perception and concludes with a computer-understandable strategy for automatic sketch map alignment. The results of the experiment shall contribute the cognitive

insights to develop GI systems for sketch map comparison guided by cognitive principles.

2 Sketch Map Alignment

The empirical study on similarity perception is focused on four sketched aspects: street networks, topological relations, directional relations, and order relations. This section explains why these four sketched aspects are important and have been investigated in this study. Moreover, ingredients of sketch maps for alignment are introduced here. During the empirical study, we applied variations to these sketch map ingredients to vary the stimuli systematically.

2.1 Accuracy and Errors in Sketch Maps

Usually, sketch maps do not represent the reality metrically-correct; rather, sketch maps correctly depict topological relations between the sketched objects. Egenhofer and Mark (1995) claimed that in human cognition "topology matters, metric refines." As external representations of cognitive maps, sketch maps are by all means distorted, rearranged, and illogical (Lynch 1960). Based on the previous empirical study, we found that four sketched aspects are usually "accurate," i.e., the binary relations between sketched objects are the same as the relations in the metric city map. These aspects are street networks, topological relations, directional relations, and order relations. They are relevant for sketch map alignment and the empirical study is centered on these sketched aspects.

2.2 Ingredients of a Sketch Map

The definition of ingredients of a sketch map helps to make a down-to-earth alignment strategy. Also, variations were applied to the ingredients to produce experiment stimuli in this empirical study. In Blaser (1998), sketch elements are divided into *objects*, *relations,* and *annotations*. The relevant alignment aspects that we mentioned before are based on sketch objects and the binary spatial relations between them. In detail, sketched objects refer to landmarks and street networks; spatial relations refer to topological relations, directional relations, and order relations. According to Blaser's (1998) definition that "an object is a logical instance or entity in a sketch" and "objects may enclose multiple independent sub-objects,"

street networks can be sub-divided into city blocks, street segments, and junctions. The following are the definitions of ingredients in a sketch map. Among the ingredients, landmarks, streets segments, junctions, and all binary relations are atomic elements, while city blocks are super elements that are composed of street segments and junctions.

- Landmarks. A landmark is an atomic element of a sketch map. In our study, a landmark is a two-dimensional object. Artificial landmarks, such as buildings are usually of regular square shape, while geographical landmarks such as water bodies are of irregular shape.
- Street segments. A street segment is an atomic element of a sketch map. In accordance with the definition of "path" from Lynch (1960) that a path network contains a main road, the junction angles, and branchings, a street segment can either be a segmentation of a main road in-between two nearby junctions or a branching connecting to the main road by a junction. A street network is represented as a set of *vertices* and a set of *edges* that connect pairs of vertices (see Fig. 1). Regardless if streets are sketched as one-dimensional or two-dimensional objects, a street segment is an *edge* in this context. Moreover, we distinguish *simple* and *complex* street segments. A simple street segment is a straight edge while a complex street segment is a curved edge. The distinction of *simple* and *complex* street segments helps for the further directional calculation using a curved street segment as a reference object.
- Junctions. A junction is a place where two or more street segments meet. Together with street segments, they form a street network. In the context of sketch map alignment, junctions are represented as a collection of *vertices*. In Figure 1, the black points are junctions and the grey ones are *end nodes*, which either indicate the boundary of a sketch map or represent dead-end streets.
- City blocks. A city block is the smallest two-dimensional area that is surrounded by street segments. City blocks form the basic unit of a city's urban fabric and most cities are composed of city blocks. They provide the space to locate buildings, e.g., the church is inside the city block surrounded by the high way and the village road; regularly arranged city blocks can be used as a distance calculation unit, e.g., the church is two blocks from here. City blocks appear quite often in sketches. Compared to street segments and junctions, this super sketch element provides more possibilities for alignment. In this study, we distinguish two types of city blocks: *closed* ones and *open* ones (see Fig. 1). The open city blocks are common at the boundary of a sketch map. We assume that people did not complete them because those streets are not necessary for the sketching task.

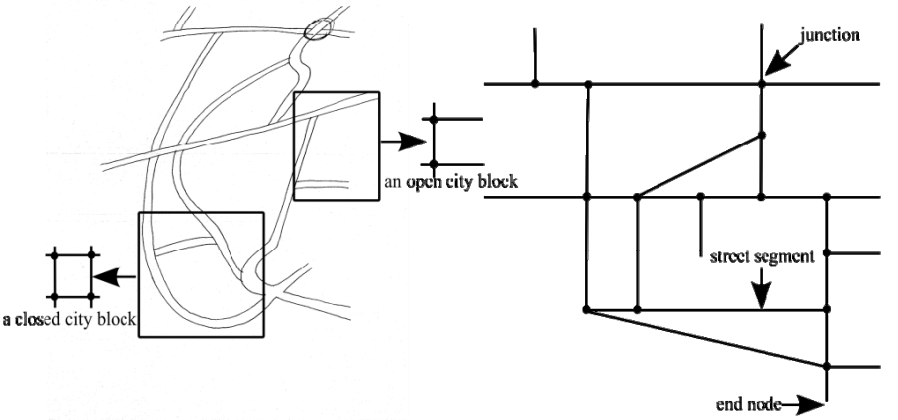


Fig. 1. A street network extracted from a sketch map used in our study and its representation using graph theory

The binary spatial relations, such as topological relations, directional relations, and order relations, are calculated in-between atomic elements, such as landmarks and street segments, as well as super sketch elements like city blocks. Annotations cannot be used for graphical alignment. Thus, they are excluded in this study.

3 Methodology

We aim to develop an automatic comparison for computers to align sketch maps and metric maps. It is necessary to investigate the impact from distortions on alignment. The empirical study is designed as a similarity-ranking experiment. During the experiment, we provided subjects with a set of sketch maps that have variations on sketch map ingredients. We distinguished four types of variations, i.e., variations applied on street networks, topological relations, directional relations, and order relations. We asked the subjects to rank these sketch maps according to their similarity to a reference map. The aim of this experiment is to investigate which stimuli are considered most similar or dissimilar, i.e., which variations are considered least severe by similarity perception and which variations are considered the most severe ones. We hypothesize that less severe variations in sketch maps are distortions that people often do when they draw sketch maps. However, variations that lead to very dissimilar sketch maps are distortions that usually do not happen when people produce sketch maps. Therefore, perceptually least severe variations do not

have negative influence and can be used by computers for automatic alignment. On the other hand, perceptually severe variations are indicators for an automatic misalignment. The similarity-ranking experiment is to test this assumption in more detail. We aim to understand whether different kinds of variations on the four sketched aspects have the same effect on similarity perception.

3.1 Materials

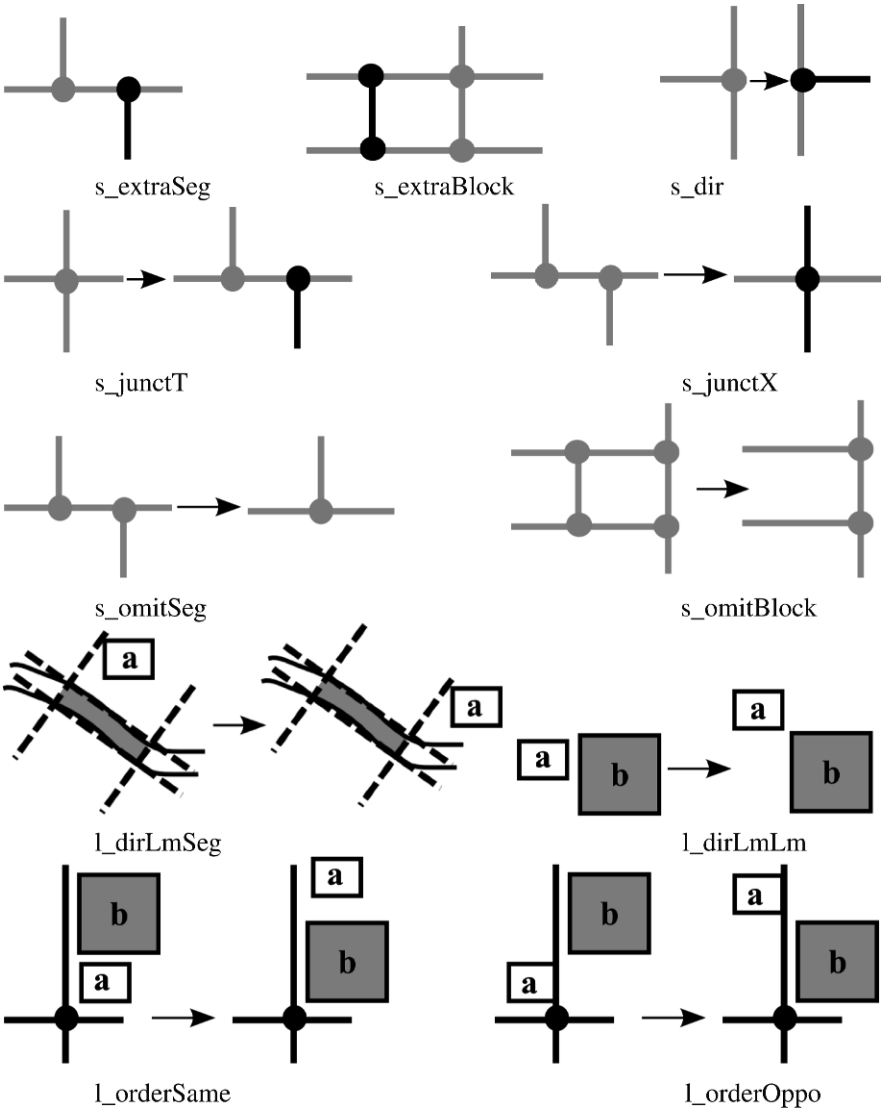
Two sketch maps depicting a part of Brueggen in Germany served as the basis for stimuli and scenario creation. Both maps were drawn in a survey perspective by people who are familiar with this area. We produced different stimuli by varying systematically the original sketch maps with different types of variations. We designed 15 types of variations. Table 1 shows the details of labels and descriptions of these variations.

Table 1. Textual labels and descriptions of 15 types of variations

Label	Description
s_extraSeg	add an extra street segment
s_extraBlock	add an extra street segment and form a city block
s_dir	alter the direction of the connected street segment to the junction
s_junctT	alter junction type from X to T
s_junctX	alter junction type from T to X
s_omitSeg	leave out a street segment
s_omitBlock	leave out a street segment and open a city block
l_dirLmSeg	alter the directional relation between a landmark and its neighboring street segment
l_dirLmLm	alter the directional relation between landmarks
l_orderSame	alter the order relation with respect to the landmark on the same side of a street
l_orderOpp	alter the order relation with respect to the landmark on the opposite side of a street
l_topoBlock	move a landmark out of a city block
l_topoSeg	alter the location of the landmark with respect to its neighboring street segments
l_touchLm	change from relation “touch” to “disjoint” with respect to a landmark
l_touchSeg	change from relation “touch” to “disjoint” with respect to a street segment

For each variation type, we produced three different examples, i.e. three stimuli with the same variation type form one *equivalence class*. Equivalence class is used for testing the consistency of perception

similarity. Seven variations ("s_") are related to street network and the remaining eight variations are related to landmarks ("l_"). In each scenario, we have 15 equivalence classes and in total 15×3 stimuli. By testing two different sketch maps with the same types of variations, negative impact from conceptual effects is decreased. Both sketch maps contain different landmarks and the drawing styles are different. The final material is equal to an overall number of 90 sketch maps as stimuli.



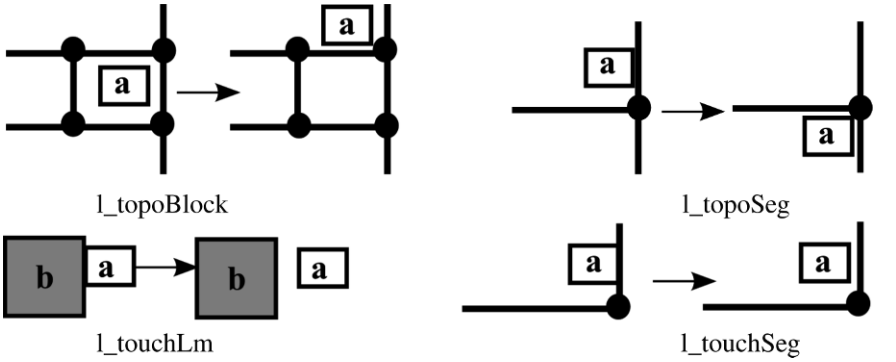


Fig. 2. Fifteen variation types on four sketched aspects to make stimuli used in the experiment

The following table shows the basic information load and ingredient types in the two scenarios. From the table, we can see that in scenario 2 the spatial layout for similarity perception is more complicated since it contains more information load.

Table 2. Information load and ingredient types used in the experiment

	Scenario 1	Scenario 2
Landmarks	**12**	**16**
With annotation	10	15
Without annotation	2	1
Street segments	**28**	**44**
Simple	22	34
Complex	6	10
Junctions*	**14**	**23**
Cross-shape	4	6
T-shape	10	17
City blocks	**12**	**18**
Open	7	10
Closed	5	8

* A junction here refers to the intersected point and its connected street segments

3.2 Subjects and Procedure

All the subjects were not familiar with the sketched areas tested in the experiment. Ten subjects joined the pre-test to assess the understandability and the feasibility of the experiment. Only minor modifications of variations were necessary. We deleted the variations on the street segments that are located at the boundary of a sketch map, because they were out of

focus during perception. The selected group of 24 subjects in the formal experiment is acquainted with geographic information science. They are either students who joined the GIS class or the faculty of the Institute for Geoinformatics. The average age of the subjects is 26.6 years, including 11 female subjects and 13 male subjects. The experiment instruction is as below:

In this experiment, you have two trials of ranking tasks for different sketch maps printed as cards. In each trial, we will show you one reference map and the task is giving different ranks to all sketch maps based on "how similar these sketch maps are to the reference map." You can put a few sketch maps together during ranking if you think they are equally similar with respect to the reference map. There is no time restriction.

All 24 subjects took two trials of the experiment. They did the similarity ranking on 45 sketch maps of scenario 1 in the first trial; after a short break they continued the second trial with another 45 sketch maps of scenario 2. A total of 22 subjects' results have been returned and analyzed. The other two subjects' results were excluded from the further analysis because the subjects only assigned in total two ranks for each scenario. Hence, they conducted rather a grouping task than a ranking task during the experiment.

4 Results

The average number of groups with equal ranking is 12.33 for the first scenario and 13.09 for the second one. The minimal number of groups a subject made is three in the first scenario and four in the second scenario, whereas for both scenarios, there are subjects who made 45 ranks for 45 stimuli. In Figure 3, the ranking distribution is quite similar between both scenarios. A strong tendency to rank more stimuli with the values ranging from 0 to 0.2 can be observed in our experiment. The remaining values that are ranging from 0.2 to 1 are rather uniformly distributed. As mentioned in the methodology section, we have 45 stimuli in each scenario with 15 equivalence classes. Since the average numbers of groups formed by subjects in both scenarios are close to 15, we can argue that in general, the similarity perception in our experiment is consistent.

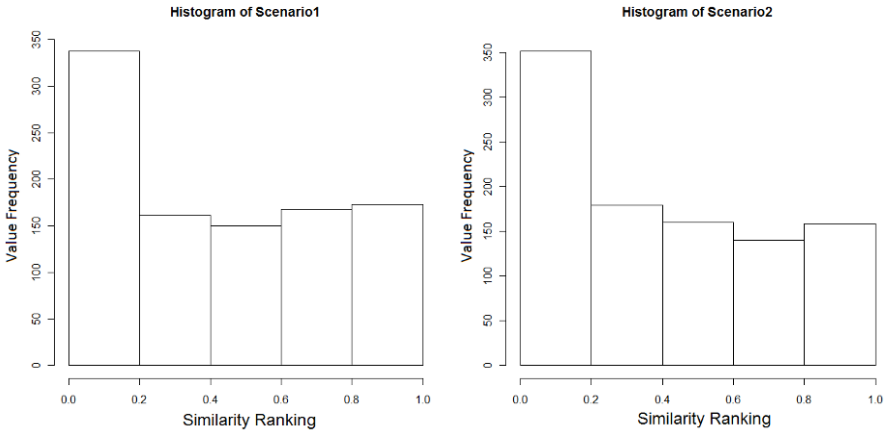


Fig. 3. Distribution of normalized similarity rankings (the range of normalized ranking value is from *0* to *1*, *0* refers to "identical" while *1* refers to the most dissimilar)

4.1 Significance of Variations

For the alignment strategy, we need information about which variations are reliable in terms of human perceptions. Reliability is assessed by investigating how the ranking values are spread among subjects. If the distribution of values for one particular variation differs significantly from all the others, we consider that variation a significant one. If not, the variation does not make a difference. The latter case applies to those variations whose distribution has a mean value around 0.5 or has a wide quartile range. Variations that are not significant are not reliable so they cannot be used for alignment.

For the comparison of ranking distributions, a Kolmogorov-Smirnov test (K-S test) was used because of its robustness when being applied to small data sets and a lack of normality for most of the distributions. The Null-Hypothesis that distributions are the same is retained on a 95% confidence level. We identify those insignificant variations with having a p-value higher than 0.05. After K-S test, the result of significant variations is shown in Table 3.

Table 3. Variations with significant distributions

Scenario	Similar	Dissimilar
1	s_omitSeg, l_dirLmSeg, l_dirLmLm l_orderOpp, l_touchSeg	s_dir, s_junctT, s_omitBlock l_topoBlock, l_topoSeg
2	s_junctT, s_junctX, l_dirLmLm l_touchLm, l_touchSeg	s_extraBlock, s_dir s_omitBlock, l_topoBlock

The most significant variation from the K-S test is *l_touchSeg* ($D = 0.4152$, $p < 10^{-8}$) in the first scenario, and *l_touchLm* ($D = 0.4576$, $p < 10^{-10}$) in the second scenario. On the other hand, we got the least significant variation *s_junctX* ($D = 0.0828$, $p = 0.7896$) in the first scenario and the least significant variation *s_extraSeg* ($D = 0.1101$, $p = 0.4413$) in the second scenario.

4.2 Consistency of Variations

In addition to the significance, we also evaluated the consistency of the 15 variations. The evaluation checks two kinds of consistency of variations: consistency across scenarios and consistency within the equivalence classes.

Apparently, the box plots of two scenarios in Figure 4 do not look the same; there seems to be scenario-dependent factors that influence particular variations. In this case, the K-S test was applied to check whether the normalized rankings of one variation from two different scenarios could be the data sample of the same distribution. This should definitely be the case if we assume that there is no difference between scenarios. We sorted variation *s_extraSeg* ($D = 0.0758$, $p = 0.9915$) out to be the most consistent while variation *s_junctT* ($D = 0.6364$, $p < 10^{-11}$) was picked as the least consistent one. For variations *s_extraSeg*, *l_touchSeg*, *s_dir*, *l_topoBlock*, *l_orderSame*, *s_omitBlock*, *s_extraBlock*, the Null-Hypothesis holds on a 95% confidence level, whereas the rest of the variations were ranked to be significantly different across scenarios and considered as inconsistent ones. One reason for this might be that these variations are perceived differently depending on the information load (see Table 2), drawing styles, or conceptual properties. Another reason could be hidden qualities that had been applied differently when creating the stimuli and caused unexpected ranking criteria. However, the inconsistent variations are still worth a closer examination. It would be interesting to know whether the inconsistency stems from the experiment design in particular or human perception in general. However, such a discussion is beyond the scope of this paper.

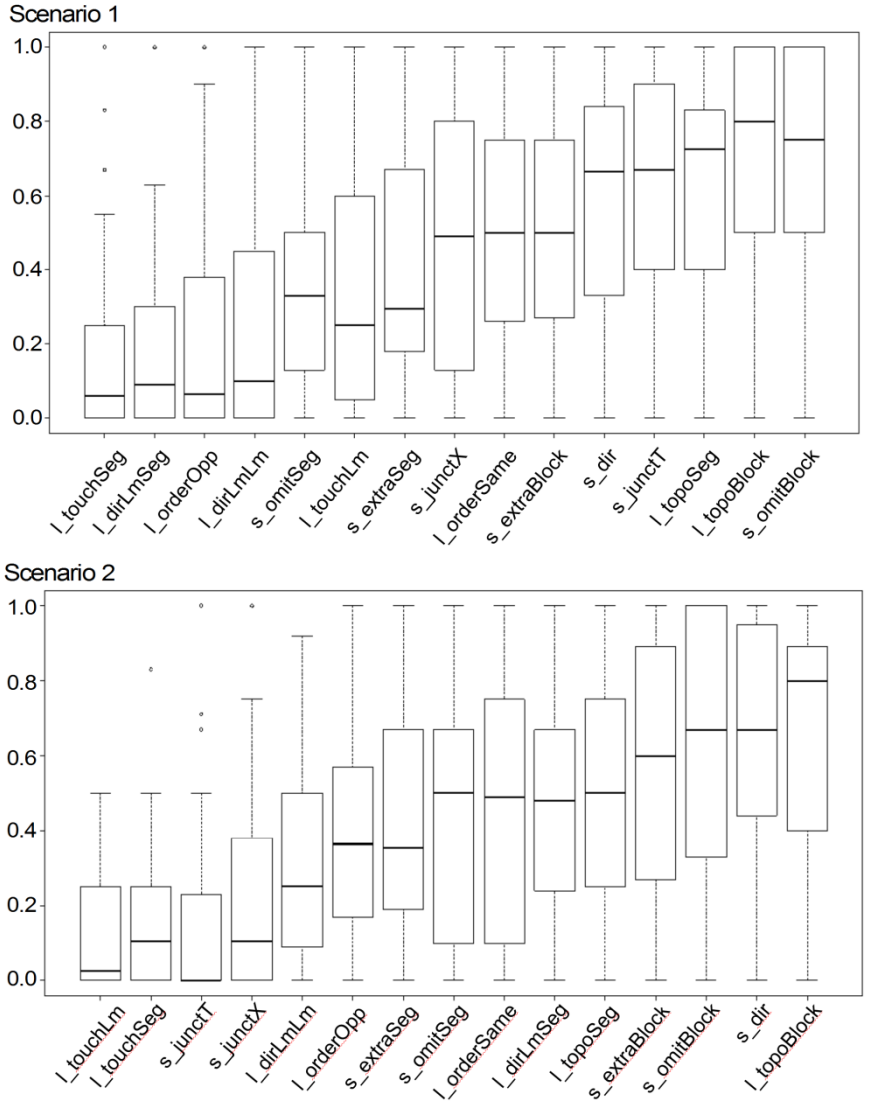


Fig. 4. Normalized similarity ranking results of two scenarios

Apart from consistency of variations across scenarios, we also checked consistency of variation with its equivalence class: distribution of normalized similarity rankings for one stimulus against the distribution of all three stimuli in the same equivalent class. Again, the K-S test was applied on a 95% confidence level. Those stimuli that do not pass K-S test are considered as outliers within the equivalence class, which means the

variation related to this equivalence class is not consistent in human perception. In fact, the p-value turned out to be quite bad for all three stimuli in that case, because for such a small set of values the distributions are highly interrelated. Variations with outliers are *l_dirLmSeg*, *l_dirLmLm,* and *l_touchLm* in the first scenario and *l_orderOppo* and *l_topoSeg* in the second one. The strongest outlier stimuli occur in variation *l_dirLmSeg* ($D = 0.3939$, $p < 0.02$) for the first scenario and in variation *l_orderOpp* ($D = 0.3485$, $p < 0.04$) for the second scenario.

4.3 Summary

Combining all findings, variations *l_touchSeg* (change from relation "touch" to "disjoint" with respect to a street segment), *s_extraBlock* (add an extra street segment and form a city block), *s_dir* (alter the direction of the connected street segment to the junction), *s_omitBlock* (leave out a street segment and open a city block), and *l_topoBlock* (move a landmark out of a city block) are the most reliable ones since they are significant within one scenario, consistent within the equivalence class, and also consistent across scenarios during similarity perception. They can be applied to the sketch map alignment depending on their ranking value; the ones that ranked as similar can be used as strategy for alignment while the dissimilar ones can be used for misalignment, e.g. a landmark touching or disconnected from the same street segment can be considered as the same (*l_touchSeg*), because people perceive these two kinds of sketched topological relations as similar. According to our assumption, while sketching a landmark near to a street, people would not distinguish these two relations. The example for misalignment strategy is: if different city block types (*s_extraBlock* and *s_omitBlock*) are detected from different maps, then it is very probable that those maps are not about the same area. According to our assumption, the perceptually dissimilar variations are usually the ones people would draw correctly.

Other variations that are only significant and consistent within one scenario (see Table 4) can still be possible candidates for sketch alignment. Further investigation of what makes such differences cross scenarios is necessary. From another perspective, combined with ingredients of alignment that we defined in section 2, the final result can be interpreted as in Table 5. Except for the variations related to binary spatial relations, the rest are variations on the street network, e.g. *s_extraBlock*, *s_junctT*, *s_junctX*, *s_omitSeg* and *s_omitBlock*, which are variations on city block, junction type and street segment.

Table 4. Final result of significance and consistency of variations across/within scenarios

Scenario	across two scenarios		within one scenario	
	Similar	Dissimilar	Similar	Dissimilar
1	l_touchSeg	s_extraBlock s_dir s_omitBlock l_topoBlock	s_omitSeg l_orderOpp	l_topoSeg
2	l_touchSeg	s_extraBlock s_dir s_omitBlock l_topoBlock	s_junctT s_junctX l_dirLmLm l_orderOpp	NULL

Table 5. Interpretation of final result with alignment ingredients and spatial relations

Alignment ingredients		Binary spatial relations between ingredients		
		Topology	Direction	Order
Landmark		l_topoBlock l_topoSeg l_touchSeg	l_dirLmLm	l_orderOpp
Street Network	Segment	l_topoSeg l_touchSeg	s_dir	l_orderOpp
	Junction*	NULL	NULL	NULL
	City block	l_topoBlock	NULL	NULL

* Junction is presented as a point object and only makes sense if it is combined with its connected street segments.

5 Discussion of Experiment

This empirical study evolved from the demand to know more about how distortions influence the alignment of sketch maps and metric maps. We are primarily interested in evidence of reliable sketched aspects that can be used for building up the alignment strategy. In the scope of this study, we designed a similarity-ranking experiment with four sketched aspects, which are street networks, topological relations, directional relations, and order relations. To make more computer-understandable elements, we defined ingredients for a sketch map and applied variations on these basic ingredients to produce the stimuli.

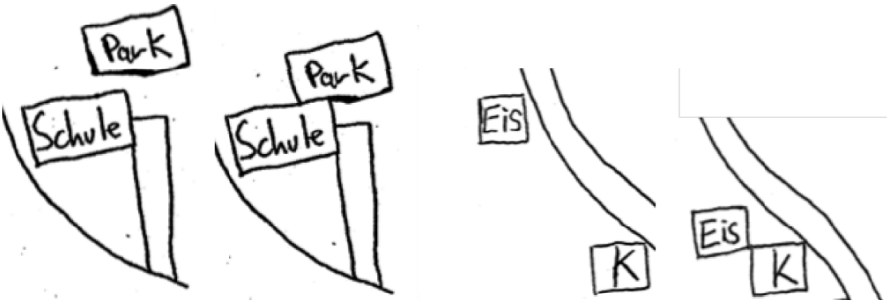
During our experiment, we experienced that subjects relied a lot on street networks while doing similarity ranking. Variations on any basic ingredients of street network were all considered severe. Among all 11

variations that are significant and consistent across scenarios or within one scenario, five of them are related to the street network (see Table 5) and six of them consist of changes of elements of the street network itself. It is not surprising that the street network plays an important role during similarity perception, because the street network is the arrangement of intersecting street segments, which is central for structuring the urban environment and human path planning. People seldom make mistakes in drawing main streets and their connectivity. On the one hand, even though minor off-street paths or minor side streets are left out in sketch maps such as *s_omitSeg*; a street network with major streets can still be used for alignment. On the other hand, different types of city blocks and junctions indicate a high possibility of different sketched areas, e.g., *s_extraBlock*, *s_dir*, *s_junctT*, *s_junctX*, *s_omitBlock,* and *l_topoSeg* can be used for the strategy for misalignment.

Spatial relations of objects in a sketch map are important for alignment as well. We have found that people were sensitive to certain changes applied to topological relations and order relations. Again, those relations are strongly related to the street network. The accuracy of topological relations between landmarks and city blocks or neighboring street segments (*l_topoBlock* and *l_topoSeg*) is an important clue to decide whether two sketch maps describe the same location or not. Order relations can also be an alignment criterion. We found that the sequence of landmarks along opposite sides of a street segment is always sketched correctly (*l_orderOpp*), which provides another possibility for alignment while no commonality exists. We can explain this result by arguing that when people are moving in a spatial environment, they usually perceive and memorize the landmarks that are located at the same side of a street.

The experiment results also reveal the observations of metric changes in sketch maps from subjects. The extreme case is the different rankings on variation *l_touchLm*, which is about the change of topological relations in-between landmarks. We changed the "touch" relation to "disjoint" relation. We assumed that all subjects would rank this variation as being very similar to the reference map since we found that people drew two landmarks being close to each other to represent the actual "touch" relation. However, we got different ranking results from different scenarios (see Fig. 4). In scenario 1, the ranking distribution is dispersed, whereas in scenario 2, the ranking distribution is concentrated and close to 0. We checked the stimuli and found that metric information affects the similarity perception. Figure 5 (left) shows that in scenario 2, the distances between two landmarks from two stimuli are very small and can still indicate a "touch" relation. However, in scenario 1 (Fig. 5 (right)), one stimulus has a much bigger distance between landmarks compared to the others, which

may indicate a "disjoint" relation instead of a "touch" relation. How metric information is represented in sketch maps is still a challenge and it can be achieved by investigating sketching habits and patterns. The results will be helpful for the alignment strategy to deal with fuzzy binary relations.



(a) (b)

Fig. 5. Metric information influences similarity ranking results: in (a), people still think these two sketches represent the same location while in (b) people think the opposite even though there is no change on objects and their annotations

6 Conclusion

Sketch map alignment is the initial step to integrate different map resources and achieve a sketching interface for VGI systems. Due to inevitable distortions in sketch maps aiming at developing an automatic comparison by computers, it is necessary to investigate how these distortions influence sketch map alignment. In this study, we focus on the alignment strategy based on spatial objects and relations, which are street networks, landmarks, topological relations, directional relations, and order relations. The empirical study shows the importance of the street network for alignment and the necessity to further distinguish its atomic elements. A city block can be extracted and defined as a super element of a street network. The type of city block as well as its topological relations with respect to other objects can be an important criterion for alignment. The correctness of order relations of objects along a street segment is also an alignment criterion. Even "*topology matters and metric refines*" holds true in most cases; metric information needs to be taken into account when building automatic alignment for computers. For future work, the definition of detailed and computational alignment ingredients are necessary and the existing calculi of topological, directional, and order

relations will be studied and revised in a sketch map context. The computer-based algorithm for alignment is expected afterwards.

Acknowledgements

The research work presented in this paper is supported by the Deutsche Forschungsgemeinschaft under grants IRTG GRK 1498 Semantic Integration of Geospatial Information. Many thanks go to all the experiment participants.

References

Beck RJ, Wood D (1976) Cognitive Transformation of Information from Urban Geographic Fields to Mental Maps. Environment and Behavior 8:199-238.

Blades M (1990) The reliability of data collected from sketch maps. Journal of Environmental Psychology 10:327-339.

Blaser A (1998) Technical report: Geo-spatial Sketches. NCGIA (TR 98-1).

Blaser A, Egenhofer M (2000) A visual tool for querying geographic databases. AVI2000-Advanced Visual Databases, Salerno, Italy.

Egenhofer M (1997) Query Processing in Spatial-Query-by-Sketch. Journal of Visual Languages and Computing 8(4): 403-424.

Egenhofer M and Mark D (1995) Naive Geography, in: Frank A and Kuhn W (eds.) COSIT '95, Austria. Lecture Notes in Computer Science, Vol. 988, Springer-Verlag, pp. 1-15.

Forbus K, Ferguson R, et al. (2001) Towards a computational model of sketching. International Conference on Intelligent User Interfaces, Sante Fe, New Mexico.

Forbus K, Usher J, Chapman V (2003) Qualitative spatial reasoning about sketch maps. Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence, Acapulco, Mexico.

Goodchild M (2007) Citizens as Sensors: The World of Volunteered Geography. GeoJournal 49(4): 211-221.

Ladd FC (1970) Black youths view their environment: Neighborhood maps. Environment and Behavior 2: 74-99.

Lynch K (1960) The Image of the City, Cambridge. MA, MIT Press.

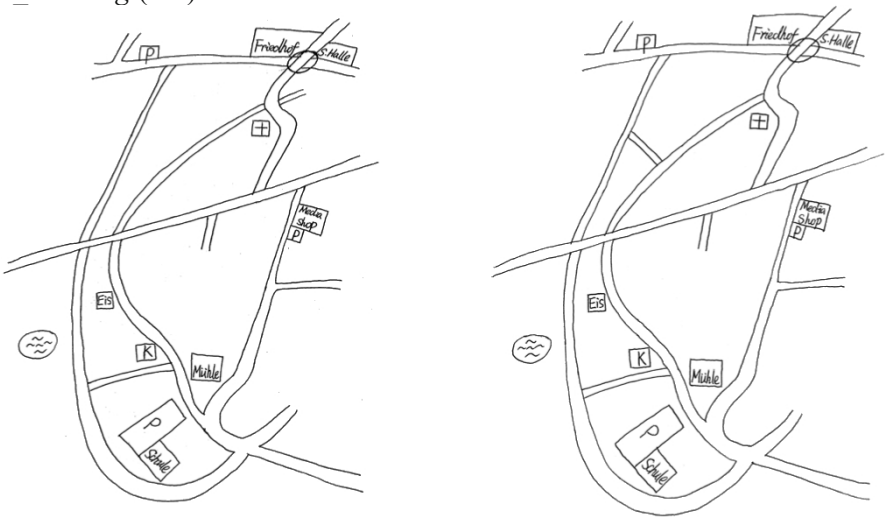
Montello, D, Freundschuh S (2005) Cognition of Geographic Information. A research agenda for geographic information science. R. B. McMaster and E. L. Usery. Boca Raton, Fl, USA, CRC Press**:** 61-91.

Tversky, B (2003) Navigating by Mind and by Body. Spatial Cognition III: Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning, Springer.

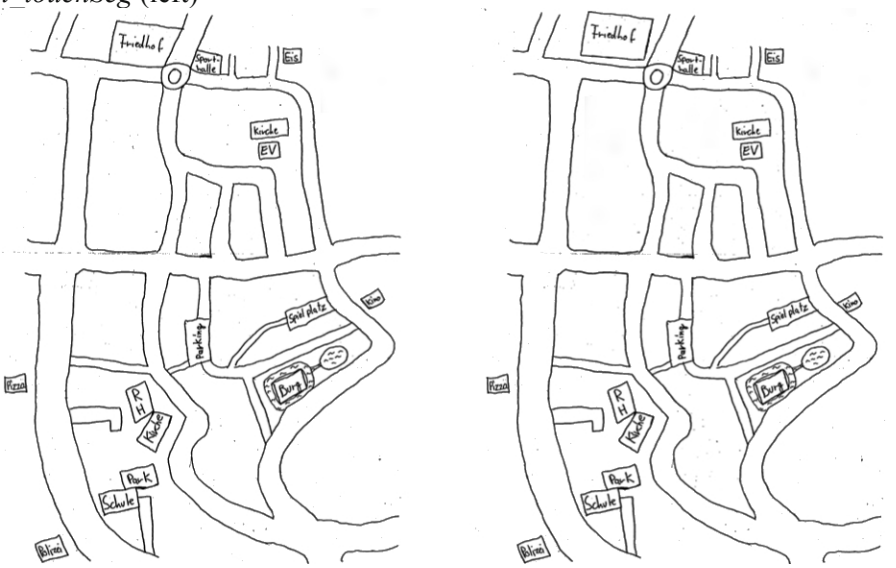
Tversky, B (2005) How to get around by mind and body - Spatial thought, spatial action. Evolution, Rationality, and Cognition: A Cognitive Science for the Twenty-First Century. A. Zilhão, Routledge**:** 135-147

Appendix 1 (reference maps and example stimuli)

Scenario 1: reference map (right) and one stimulus with variation type *s_extraSeg* (left)



Scenario 2: reference map (right) and one stimulus with variation type *l_touchSeg* (left)



Appendix 2 (tables of similarity ranking values)

Scenario 1

	scenario 1: original / normalized similarity rankings for variation 1 - 5														
	1			2			3			4			5		
p 01	3 / 0.2	3 / 0.2	3 / 0.2	6 / 0.5	6 / 0.5	6 / 0.5	7 / 0.6	7 / 0.6	7 / 0.6	10 / 0.9	10 / 0.9	10 / 0.9	8 / 0.7	8 / 0.7	10 / 0.9
p 02	7 / 0.29	7 / 0.29	7 / 0.29	7 / 0.29	13 / 0.58	13 / 0.58	8 / 0.34	8 / 0.34	8 / 0.34	11 / 0.48	11 / 0.48	9 / 0.39	5 / 0.2	10 / 0.43	11 / 0.48
p 03	27 / 0.6	31 / 0.69	39 / 0.87	29 / 0.64	44 / 0.98	43 / 0.96	35 / 0.78	30 / 0.66	38 / 0.85	36 / 0.8	45 / 1	40 / 0.89	28 / 0.62	41 / 0.91	42 / 0.94
p 04	5 / 0.8	1 / 0	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8	6 / 1	6 / 1	5 / 0.8	6 / 1	6 / 1	6 / 1	6 / 1	6 / 1	6 / 1
p 05	1 / 0	3 / 0.67	3 / 0.67	3 / 0.67	3 / 0.67	3 / 0.67	4 / 1	4 / 1	4 / 1	1 / 0	4 / 1	4 / 1	4 / 1	1 / 0	2 / 0.34
p 06	6 / 0.63	1 / 0	7 / 0.75	8 / 0.88	6 / 0.63	7 / 0.75	3 / 0.25	7 / 0.75	8 / 0.88	6 / 0.63	3 / 0.25	8 / 0.88	1 / 0	2 / 0.13	4 / 0.38
p 07	2 / 0.17	1 / 0	3 / 0.34	1 / 0	3 / 0.34	3 / 0.34	2 / 0.17	1 / 0	2 / 0.17	5 / 0.67	4 / 0.5	2 / 0.17	1 / 0	3 / 0.34	1 / 0
p 08	2 / 0.34	1 / 0	1 / 0	2 / 0.34	2 / 0.34	2 / 0.34	3 / 0.67	2 / 0.34	3 / 0.67	3 / 0.67	3 / 0.67	3 / 0.67	2 / 0.34	2 / 0.34	1 / 0
p 09	2 / 0.2	2 / 0.2	3 / 0.4	2 / 0.2	2 / 0.2	2 / 0.2	6 / 1	2 / 0.2	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8
p 10	14 / 0.3	21 / 0.46	15 / 0.32	18 / 0.39	17 / 0.37	16 / 0.35	35 / 0.78	31 / 0.69	25 / 0.55	30 / 0.66	40 / 0.89	26 / 0.57	24 / 0.53	37 / 0.82	39 / 0.87
p 11	4 / 0.6	4 / 0.6	4 / 0.6	4 / 0.6	4 / 0.6	3 / 0.4	4 / 0.6	4 / 0.6	4 / 0.6	1 / 0	1 / 0	4 / 0.6	4 / 0.6	6 / 1	1 / 0
p 12	2 / 0.25	1 / 0	1 / 0	2 / 0.25	2 / 0.25	3 / 0.5	5 / 1	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5
p 13	6 / 0.17	16 / 0.49	16 / 0.49	8 / 0.23	22 / 0.68	21 / 0.65	19 / 0.59	14 / 0.42	23 / 0.71	30 / 0.94	24 / 0.75	26 / 0.81	23 / 0.71	31 / 0.97	10 / 0.3
p 14	2 / 0.13	2 / 0.13	2 / 0.13	2 / 0.13	2 / 0.13	2 / 0.13	8 / 0.88	8 / 0.88	8 / 0.88	6 / 0.63	9 / 1	6 / 0.63	6 / 0.63	9 / 1	4 / 0.38
p 15	3 / 0.19	3 / 0.19	3 / 0.19	3 / 0.19	4 / 0.28	4 / 0.28	3 / 0.19	3 / 0.19	3 / 0.19	2 / 0.1	2 / 0.1	9 / 0.73	2 / 0.1	2 / 0.1	7 / 0.55
p 16	7 / 0.55	1 / 0	12 / 1	12 / 1	9 / 0.73	9 / 0.73	11 / 0.91	8 / 0.64	6 / 0.46	10 / 0.82	1 / 0	6 / 0.46	2 / 0.1	2 / 0.1	1 / 0
p 17	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	1 / 0	2 / 0.25	4 / 0.75	4 / 0.75	4 / 0.75	1 / 0	1 / 0	4 / 0.75	4 / 0.75	4 / 0.75	1 / 0
p 18	7 / 1	7 / 1	7 / 1	7 / 1	7 / 1	7 / 1	7 / 1	2 / 0.17	2 / 0.17	2 / 0.17	2 / 0.17	2 / 0.17	2 / 0.17	3 / 0.34	2 / 0.17
p 19	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	5 / 1	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	1 / 0
p 20	8 / 0.31	7 / 0.27	13 / 0.53	14 / 0.57	12 / 0.48	17 / 0.7	18 / 0.74	11 / 0.44	7 / 0.27	23 / 0.96	21 / 0.87	7 / 0.27	15 / 0.61	4 / 0.14	1 / 0
p 21	1 / 0	3 / 0.2	2 / 0.1	3 / 0.2	6 / 0.5	3 / 0.2	2 / 0.1	2 / 0.1	2 / 0.1	5 / 0.4	5 / 0.4	11 / 1	3 / 0.2	5 / 0.4	5 / 0.4
p 22	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	3 / 1	1 / 0

	scenario 1: original / normalized similarity rankings for variation 6 - 10														
	6			7			8			9			10		
p 01	5 / 0.4	5 / 0.4	5 / 0.4	11 / 1	11 / 1	11 / 1	2 / 0.1	2 / 0.1	4 / 0.3	2 / 0.1	2 / 0.1	4 / 0.3	9 / 0.8	9 / 0.8	9 / 0.8
p 02	1 / 0	6 / 0.24	6 / 0.24	16 / 0.72	12 / 0.53	12 / 0.53	1 / 0	2 / 0.05	4 / 0.15	1 / 0	1 / 0	4 / 0.15	21 / 0.96	17 / 0.77	15 / 0.67
p 03	14 / 0.3	16 / 0.35	17 / 0.37	33 / 0.73	15 / 0.32	34 / 0.75	5 / 0.1	9 / 0.19	24 / 0.53	8 / 0.16	10 / 0.21	21 / 0.46	23 / 0.5	26 / 0.57	25 / 0.55
p 04	4 / 0.6	4 / 0.6	4 / 0.6	4 / 0.6	4 / 0.6	4 / 0.6	1 / 0	2 / 0.2	2 / 0.2	1 / 0	2 / 0.2	2 / 0.2	1 / 0	2 / 0.2	1 / 0
p 05	3 / 0.67	3 / 0.67	3 / 0.67	3 / 0.67	3 / 0.67	3 / 0.67	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.34	2 / 0.34	2 / 0.34	1 / 0
p 06	3 / 0.25	2 / 0.13	1 / 0	9 / 1	6 / 0.63	7 / 0.75	2 / 0.13	4 / 0.38	4 / 0.38	3 / 0.25	2 / 0.13	6 / 0.63	5 / 0.5	4 / 0.38	2 / 0.13
p 07	1 / 0	1 / 0	2 / 0.17	6 / 0.84	1 / 0	7 / 1	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 08	2 / 0.34	1 / 0	2 / 0.34	2 / 0.34	2 / 0.34	2 / 0.34	1 / 0	1 / 0	4 / 1	1 / 0	4 / 1	4 / 1	3 / 0.67	4 / 1	3 / 0.67
p 09	3 / 0.4	5 / 0.8	3 / 0.4	6 / 1	6 / 1	6 / 1	1 / 0	3 / 0.4	3 / 0.4	1 / 0	1 / 0	4 / 0.6	5 / 0.8	1 / 0	2 / 0.2
p 10	3 / 0.05	29 / 0.64	19 / 0.41	45 / 1	33 / 0.73	44 / 0.98	2 / 0.03	12 / 0.25	13 / 0.28	20 / 0.44	9 / 0.19	27 / 0.6	38 / 0.85	22 / 0.48	23 / 0.5
p 11	1 / 0	1 / 0	4 / 0.6	6 / 1	1 / 0	6 / 1	1 / 0	2 / 0.2	2 / 0.2	1 / 0	1 / 0	4 / 0.6	3 / 0.4	5 / 0.8	1 / 0
p 12	1 / 0	3 / 0.5	2 / 0.25	5 / 1	4 / 0.75	5 / 1	1 / 0	1 / 0	3 / 0.5	1 / 0	1 / 0	2 / 0.25	5 / 1	4 / 0.75	4 / 0.75
p 13	15 / 0.46	18 / 0.55	11 / 0.33	32 / 1	29 / 0.91	32 / 1	2 / 0.04	3 / 0.07	12 / 0.36	1 / 0	3 / 0.07	12 / 0.36	9 / 0.26	17 / 0.52	20 / 0.62
p 14	5 / 0.5	1 / 0	1 / 0	5 / 0.5	5 / 0.5	5 / 0.5	3 / 0.25	4 / 0.38	6 / 0.63	1 / 0	1 / 0	6 / 0.63	7 / 0.75	7 / 0.75	7 / 0.75
p 15	3 / 0.19	3 / 0.19	3 / 0.19	12 / 1	11 / 0.91	11 / 0.91	1 / 0	7 / 0.55	7 / 0.55	7 / 0.55	7 / 0.55	7 / 0.55	8 / 0.64	6 / 0.46	6 / 0.46
p 16	7 / 0.55	3 / 0.19	8 / 0.64	11 / 0.91	4 / 0.28	5 / 0.37	2 / 0.1	1 / 0	2 / 0.1	1 / 0	1 / 0	1 / 0	7 / 0.55	2 / 0.1	2 / 0.1
p 17	1 / 0	3 / 0.5	1 / 0	3 / 0.5	3 / 0.5	3 / 0.5	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	4 / 0.75	4 / 0.75	4 / 0.75
p 18	3 / 0.34	3 / 0.34	3 / 0.34	3 / 0.34	3 / 0.34	3 / 0.34	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	4 / 0.5	4 / 0.5	4 / 0.5
p 19	4 / 0.75	1 / 0	4 / 0.75	4 / 0.75	5 / 1	5 / 1	1 / 0	2 / 0.25	3 / 0.5	1 / 0	1 / 0	3 / 0.5	3 / 0.5	2 / 0.25	3 / 0.5
p 20	9 / 0.35	7 / 0.27	6 / 0.22	22 / 0.92	6 / 0.22	24 / 1	2 / 0.05	5 / 0.18	6 / 0.22	4 / 0.14	1 / 0	7 / 0.27	9 / 0.35	7 / 0.27	6 / 0.22
p 21	1 / 0	3 / 0.2	3 / 0.2	10 / 0.9	10 / 0.9	6 / 0.5	1 / 0	7 / 0.6	7 / 0.6	7 / 0.6	9 / 0.8	7 / 0.6	1 / 0	8 / 0.7	8 / 0.7
p 22	3 / 1	1 / 0	3 / 1	3 / 1	3 / 1	3 / 1	1 / 0	1 / 0	2 / 0.5	1 / 0	2 / 0.5	2 / 0.5	2 / 0.5	2 / 0.5	2 / 0.5

	scenario 1: original / normalized similarity rankings for variation 11 - 15														
	11			12			13			14			15		
p 01	2 / 0.1	4 / 0.3	2 / 0.1	9 / 0.8	9 / 0.8	9 / 0.8	9 / 0.8	9 / 0.8	9 / 0.8	1 / 0	2 / 0.1	2 / 0.1	1 / 0	1 / 0	1 / 0
p 02	1 / 0	4 / 0.15	2 / 0.05	20 / 0.91	19 / 0.86	19 / 0.86	22 / 1	14 / 0.62	18 / 0.81	3 / 0.1	1 / 0	3 / 0.1	3 / 0.1	3 / 0.1	3 / 0.1
p 03	11 / 0.23	7 / 0.14	13 / 0.28	18 / 0.39	20 / 0.44	22 / 0.48	32 / 0.71	12 / 0.25	37 / 0.82	3 / 0.05	4 / 0.07	19 / 0.41	1 / 0	2 / 0.03	6 / 0.12
p 04	1 / 0	2 / 0.2	1 / 0	3 / 0.4	3 / 0.4	3 / 0.4	3 / 0.4	3 / 0.4	3 / 0.4	1 / 0	1 / 0	2 / 0.2	1 / 0	1 / 0	1 / 0
p 05	1 / 0	2 / 0.34	2 / 0.34	1 / 0	2 / 0.34	2 / 0.34	2 / 0.34	2 / 0.34	2 / 0.34	1 / 0	1 / 0	2 / 0.34	1 / 0	1 / 0	1 / 0
p 06	1 / 0	2 / 0.13	5 / 0.5	5 / 0.63	7 / 0.75	6 / 0.63	6 / 0.63	4 / 0.38	8 / 0.88	5 / 0.5	2 / 0.13	6 / 0.63	3 / 0.25	3 / 0.25	4 / 0.38
p 07	1 / 0	1 / 0	1 / 0	2 / 0.17	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.17	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 08	1 / 0	4 / 1	1 / 0	3 / 0.67	4 / 1	3 / 0.67	3 / 0.67	4 / 1	3 / 0.67	4 / 1	1 / 0	4 / 1	1 / 0	1 / 0	4 / 1
p 09	1 / 0	1 / 0	1 / 0	4 / 0.6	6 / 1	4 / 0.6	6 / 1	5 / 0.8	4 / 0.6	5 / 0.8	5 / 0.8	5 / 0.8	1 / 0	1 / 0	1 / 0
p 10	4 / 0.07	11 / 0.23	10 / 0.21	41 / 0.91	43 / 0.96	34 / 0.75	42 / 0.94	28 / 0.62	36 / 0.8	7 / 0.14	5 / 0.1	32 / 0.71	6 / 0.12	1 / 0	8 / 0.16
p 11	1 / 0	3 / 0.4	1 / 0	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8	5 / 0.8	1 / 0	2 / 0.2	4 / 0.6	2 / 0.2	2 / 0.2	2 / 0.2	2 / 0.2
p 12	1 / 0	1 / 0	1 / 0	5 / 1	5 / 1	5 / 1	5 / 1	4 / 0.75	5 / 1	2 / 0.25	1 / 0	5 / 1	1 / 0	1 / 0	1 / 0
p 13	3 / 0.07	13 / 0.39	3 / 0.07	25 / 0.78	25 / 0.78	25 / 0.78	26 / 0.81	19 / 0.59	28 / 0.88	7 / 0.2	5 / 0.13	27 / 0.84	4 / 0.1	4 / 0.1	1 / 0
p 14	4 / 0.38	6 / 0.63	4 / 0.38	9 / 1	9 / 1	9 / 1	9 / 1	6 / 0.63	9 / 1	4 / 0.38	1 / 0	6 / 0.63	4 / 0.38	1 / 0	1 / 0
p 15	7 / 0.55	7 / 0.55	7 / 0.55	10 / 0.82	10 / 0.82	10 / 0.82	10 / 0.82	7 / 0.55	10 / 0.82	5 / 0.37	5 / 0.37	7 / 0.55	5 / 0.37	5 / 0.37	5 / 0.37
p 16	1 / 0	1 / 0	6 / 0.46	8 / 0.64	7 / 0.55	7 / 0.55	12 / 1	2 / 0.1	3 / 0.19	6 / 0.46	8 / 0.64	2 / 0.1	1 / 0	7 / 0.55	6 / 0.46
p 17	1 / 0	1 / 0	1 / 0	5 / 1	5 / 1	5 / 1	5 / 1	4 / 0.75	5 / 1	2 / 0.25	1 / 0	4 / 0.75	1 / 0	1 / 0	1 / 0
p 18	1 / 0	1 / 0	1 / 0	6 / 0.84	6 / 0.84	6 / 0.84	6 / 0.84	1 / 0	6 / 0.84	5 / 0.67	5 / 0.67	1 / 0	5 / 0.67	6 / 0.84	6 / 0.84
p 19	1 / 0	3 / 0.5	2 / 0.25	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	2 / 0.25	1 / 0	3 / 0.5	2 / 0.25	2 / 0.25	2 / 0.25
p 20	2 / 0.05	1 / 0	6 / 0.22	9 / 0.35	9 / 0.35	16 / 0.66	20 / 0.83	4 / 0.14	19 / 0.79	9 / 0.35	3 / 0.09	10 / 0.4	6 / 0.22	4 / 0.14	6 / 0.22
p 21	1 / 0	10 / 0.9	9 / 0.8	11 / 1	11 / 1	11 / 1	11 / 1	7 / 0.6	8 / 0.7	7 / 0.6	7 / 0.6	8 / 0.7	4 / 0.3	1 / 0	4 / 0.3
p 22	2 / 0.5	2 / 0.5	2 / 0.5	3 / 1	3 / 1	3 / 1	1 / 0	2 / 0.5	3 / 1	2 / 0.5	1 / 0	3 / 1	1 / 0	2 / 0.5	1 / 0

Scenario 2

	scenario 2: original / normalized similarity rankings for variation 1 - 5														
	1			2			3			4			5		
p 01	3 / 0.23	3 / 0.23	3 / 0.23	9 / 0.89	9 / 0.89	9 / 0.89	5 / 0.45	5 / 0.45	5 / 0.45	7 / 0.67	7 / 0.67	7 / 0.67	7 / 0.67	7 / 0.67	7 / 0.67
p 02	8 / 0.19	8 / 0.19	8 / 0.19	12 / 0.3	12 / 0.3	11 / 0.28	21 / 0.55	14 / 0.36	20 / 0.52	1 / 0	1 / 0	5 / 0.11	5 / 0.11	5 / 0.11	2 / 0.03
p 03	31 / 0.69	33 / 0.73	32 / 0.71	43 / 0.96	36 / 0.8	35 / 0.78	34 / 0.75	44 / 0.98	45 / 1	3 / 0.05	1 / 0	2 / 0.03	27 / 0.6	13 / 0.28	4 / 0.07
p 04	5 / 0.67	5 / 0.67	5 / 0.67	5 / 0.67	5 / 0.67	4 / 0.5	6 / 0.84	6 / 0.84	6 / 0.84	1 / 0	7 / 1	7 / 1	1 / 0	7 / 1	7 / 1
p 05	2 / 0.25	2 / 0.25	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	5 / 1	5 / 1	5 / 1	1 / 0	1 / 0	5 / 1	5 / 1	1 / 0	5 / 1
p 06	4 / 0.75	4 / 0.75	4 / 0.75	4 / 0.75	5 / 1	5 / 1	5 / 1	4 / 0.75	5 / 1	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 07	2 / 0.2	4 / 0.6	4 / 0.6	1 / 0	3 / 0.4	5 / 0.8	3 / 0.4	4 / 0.6	3 / 0.4	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.2
p 08	4 / 1	1 / 0	4 / 1	4 / 1	4 / 1	4 / 1	4 / 1	4 / 1	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.34	1 / 0
p 09	2 / 0.25	2 / 0.25	1 / 0	1 / 0	5 / 1	1 / 0	4 / 0.75	1 / 0	4 / 0.75	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.25
p 10	18 / 0.39	15 / 0.32	14 / 0.3	25 / 0.55	21 / 0.46	20 / 0.44	24 / 0.53	19 / 0.41	30 / 0.66	12 / 0.25	1 / 0	11 / 0.23	26 / 0.57	22 / 0.48	2 / 0.03
p 11	4 / 0.34	5 / 0.45	4 / 0.34	8 / 0.78	7 / 0.67	9 / 0.89	5 / 0.45	5 / 0.45	5 / 0.45	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 12	1 / 0	1 / 0	1 / 0	5 / 1	5 / 1	5 / 1	3 / 0.5	4 / 0.75	2 / 0.25	3 / 0.5	1 / 0	1 / 0	2 / 0.25	3 / 0.5	1 / 0
p 13	14 / 0.55	14 / 0.55	14 / 0.55	15 / 0.59	15 / 0.59	19 / 0.75	16 / 0.63	16 / 0.63	16 / 0.63	10 / 0.38	10 / 0.38	10 / 0.38	10 / 0.38	10 / 0.38	2 / 0.05
p 14	4 / 0.43	4 / 0.43	4 / 0.43	4 / 0.43	4 / 0.43	4 / 0.43	7 / 0.86	7 / 0.86	7 / 0.86	1 / 0	1 / 0	6 / 0.72	1 / 0	6 / 0.72	2 / 0.15
p 15	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12	3 / 0.23	8 / 0.78	2 / 0.12	8 / 0.78	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 16	7 / 0.47	7 / 0.47	1 / 0	1 / 0	9 / 0.62	4 / 0.24	9 / 0.62	13 / 0.93	8 / 0.54	1 / 0	14 / 1	1 / 0	4 / 0.24	1 / 0	10 / 0.7
p 17	2 / 0.25	2 / 0.25	1 / 0	1 / 0	2 / 0.25	2 / 0.25	5 / 1	1 / 0	5 / 1	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 18	7 / 1	7 / 1	7 / 1	7 / 1	7 / 1	7 / 1	5 / 0.67	5 / 0.67	7 / 1	2 / 0.17	1 / 0	1 / 0	1 / 0	2 / 0.17	1 / 0
p 19	5 / 1	5 / 1	5 / 1	4 / 0.75	5 / 1	4 / 0.75	4 / 0.75	5 / 1	5 / 1	1 / 0	1 / 0	1 / 0	1 / 0	4 / 0.75	3 / 0.5
p 20	9 / 0.39	11 / 0.48	10 / 0.43	1 / 0	5 / 0.2	5 / 0.2	21 / 0.96	14 / 0.62	5 / 0.2	3 / 0.1	3 / 0.1	3 / 0.1	3 / 0.1	3 / 0.1	1 / 0
p 21	1 / 0	1 / 0	1 / 0	4 / 0.34	3 / 0.23	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12
p 23	2 / 0.15	2 / 0.15	2 / 0.15	5 / 0.58	5 / 0.58	5 / 0.58	2 / 0.15	8 / 1	8 / 1	3 / 0.29	3 / 0.29	3 / 0.29	3 / 0.29	3 / 0.29	3 / 0.29

	scenario 2: original / normalized similarity rankings for variation 6 - 10														
	6			7			8			9			10		
p 01	6 / 0.56	10 / 1	6 / 0.56	3 / 0.23	10 / 1	10 / 1	4 / 0.34	4 / 0.34	8 / 0.78	2 / 0.12	4 / 0.34	2 / 0.12	8 / 0.78	8 / 0.78	8 / 0.78
p 02	9 / 0.22	6 / 0.14	7 / 0.17	13 / 0.33	13 / 0.33	10 / 0.25	18 / 0.46	22 / 0.57	23 / 0.6	19 / 0.49	3 / 0.06	4 / 0.09	26 / 0.68	38 / 1	27 / 0.71
p 03	28 / 0.62	5 / 0.1	29 / 0.64	25 / 0.55	24 / 0.53	30 / 0.66	16 / 0.35	40 / 0.89	41 / 0.91	18 / 0.39	17 / 0.37	19 / 0.41	38 / 0.85	22 / 0.48	20 / 0.44
p 04	4 / 0.5	1 / 0	4 / 0.5	4 / 0.5	4 / 0.5	4 / 0.5	2 / 0.17	2 / 0.17	2 / 0.17	2 / 0.17	1 / 0	2 / 0.17	2 / 0.17	2 / 0.17	1 / 0
p 05	4 / 0.75	1 / 0	2 / 0.25	4 / 0.75	4 / 0.75	4 / 0.75	3 / 0.5	3 / 0.5	3 / 0.5	2 / 0.25	1 / 0	2 / 0.25	1 / 0	3 / 0.5	3 / 0.5
p 06	4 / 0.75	1 / 0	4 / 0.75	5 / 1	5 / 1	5 / 1	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	1 / 0	2 / 0.25	3 / 0.5	3 / 0.5	3 / 0.5
p 07	2 / 0.2	1 / 0	3 / 0.4	6 / 1	3 / 0.4	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 08	4 / 1	1 / 0	1 / 0	1 / 0	4 / 1	1 / 0	2 / 0.34	2 / 0.34	3 / 0.67	1 / 0	2 / 0.34	2 / 0.34	3 / 0.67	3 / 0.67	3 / 0.67
p 09	1 / 0	1 / 0	4 / 0.75	1 / 0	5 / 1	3 / 0.5	1 / 0	3 / 0.5	4 / 0.75	2 / 0.25	1 / 0	2 / 0.25	4 / 0.75	1 / 0	4 / 0.75
p 10	29 / 0.64	17 / 0.37	23 / 0.5	27 / 0.6	36 / 0.8	28 / 0.62	31 / 0.69	38 / 0.85	43 / 0.96	33 / 0.73	5 / 0.1	32 / 0.71	44 / 0.98	41 / 0.91	3 / 0.05
p 11	1 / 0	1 / 0	5 / 0.45	10 / 1	10 / 1	1 / 0	3 / 0.23	4 / 0.34	4 / 0.34	3 / 0.23	3 / 0.23	2 / 0.12	1 / 0	2 / 0.12	2 / 0.12
p 12	2 / 0.25	3 / 0.5	3 / 0.5	5 / 1	5 / 1	4 / 0.75	3 / 0.5	4 / 0.75	5 / 1	3 / 0.5	4 / 0.75	3 / 0.5	1 / 0	4 / 0.75	4 / 0.75
p 13	13 / 0.5	13 / 0.5	18 / 0.71	20 / 0.8	18 / 0.71	18 / 0.71	4 / 0.13	21 / 0.84	21 / 0.84	11 / 0.42	6 / 0.21	11 / 0.42	1 / 0	11 / 0.42	9 / 0.34
p 14	5 / 0.58	5 / 0.58	5 / 0.58	5 / 0.58	5 / 0.58	5 / 0.58	2 / 0.15	2 / 0.15	3 / 0.29	2 / 0.15	2 / 0.15	2 / 0.15	7 / 0.86	3 / 0.29	3 / 0.29
p 15	10 / 1	1 / 0	10 / 1	10 / 1	10 / 1	10 / 1	6 / 0.56	6 / 0.56	7 / 0.67	6 / 0.56	6 / 0.56	6 / 0.56	5 / 0.45	5 / 0.45	5 / 0.45
p 16	1 / 0	12 / 0.85	8 / 0.54	1 / 0	4 / 0.24	6 / 0.39	12 / 0.85	2 / 0.08	2 / 0.08	1 / 0	1 / 0	1 / 0	1 / 0	11 / 0.77	10 / 0.7
p 17	4 / 0.75	1 / 0	4 / 0.75	1 / 0	4 / 0.75	1 / 0	1 / 0	3 / 0.5	3 / 0.5	1 / 0	1 / 0	1 / 0	5 / 1	3 / 0.5	3 / 0.5
p 18	5 / 0.67	5 / 0.67	5 / 0.67	5 / 0.67	5 / 0.67	5 / 0.67	4 / 0.5	4 / 0.5	4 / 0.5	4 / 0.5	4 / 0.5	4 / 0.5	6 / 0.84	6 / 0.84	6 / 0.84
p 19	4 / 0.75	1 / 0	1 / 0	5 / 1	5 / 1	4 / 0.75	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	1 / 0
p 20	12 / 0.53	13 / 0.58	3 / 0.1	22 / 1	20 / 0.91	5 / 0.2	3 / 0.1	6 / 0.24	18 / 0.81	8 / 0.34	3 / 0.1	8 / 0.34	3 / 0.1	3 / 0.1	5 / 0.2
p 21	3 / 0.23	3 / 0.23	3 / 0.23	4 / 0.34	3 / 0.23	3 / 0.23	7 / 0.67	5 / 0.45	10 / 1	6 / 0.56	7 / 0.67	6 / 0.56	8 / 0.78	8 / 0.78	8 / 0.78
p 23	6 / 0.72	6 / 0.72	4 / 0.43	6 / 0.72	6 / 0.72	4 / 0.43	1 / 0	7 / 0.86	7 / 0.86	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0

	scenario 2: original / normalized similarity rankings for variation 11 - 15														
	11			12			13			14			15		
p 01	8 / 0.78	2 / 0.12	2 / 0.12	8 / 0.78	8 / 0.78	8 / 0.78	8 / 0.78	2 / 0.12	4 / 0.34	2 / 0.12	1 / 0	1 / 0	2 / 0.12	1 / 0	2 / 0.12
p 02	23 / 0.6	16 / 0.41	24 / 0.63	15 / 0.38	29 / 0.76	15 / 0.38	30 / 0.79	17 / 0.44	25 / 0.65	1 / 0	1 / 0	3 / 0.06	1 / 0	3 / 0.06	3 / 0.06
p 03	42 / 0.94	26 / 0.57	15 / 0.32	23 / 0.5	39 / 0.87	37 / 0.82	12 / 0.25	14 / 0.3	11 / 0.23	9 / 0.19	7 / 0.14	21 / 0.46	10 / 0.21	8 / 0.16	6 / 0.12
p 04	2 / 0.17	2 / 0.17	2 / 0.17	3 / 0.34	3 / 0.34	3 / 0.34	3 / 0.34	2 / 0.17	2 / 0.17	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 05	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	3 / 0.5	2 / 0.25	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 06	4 / 0.75	1 / 0	2 / 0.25	4 / 0.75	5 / 1	3 / 0.5	3 / 0.5	2 / 0.25	2 / 0.25	3 / 0.5	1 / 0	1 / 0	3 / 0.5	2 / 0.25	3 / 0.5
p 07	3 / 0.4	1 / 0	1 / 0	2 / 0.2	3 / 0.4	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 08	4 / 1	1 / 0	2 / 0.34	4 / 1	3 / 0.67	4 / 1	3 / 0.67	2 / 0.34	3 / 0.67	1 / 0	2 / 0.34	2 / 0.34	2 / 0.34	2 / 0.34	1 / 0
p 09	5 / 1	2 / 0.25	2 / 0.25	4 / 0.75	5 / 1	1 / 0	4 / 0.75	3 / 0.5	3 / 0.5	1 / 0	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	1 / 0
p 10	42 / 0.94	13 / 0.28	35 / 0.78	40 / 0.89	45 / 1	37 / 0.82	39 / 0.87	16 / 0.35	34 / 0.75	9 / 0.19	4 / 0.07	6 / 0.12	8 / 0.16	7 / 0.14	10 / 0.21
p 11	6 / 0.56	1 / 0	4 / 0.34	5 / 0.56	2 / 0.12	4 / 0.34	4 / 0.34	2 / 0.12	6 / 0.56	1 / 0	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12	2 / 0.12
p 12	5 / 1	3 / 0.5	4 / 0.75	5 / 1	5 / 1	5 / 1	5 / 1	4 / 0.75	3 / 0.5	2 / 0.25	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.25
p 13	22 / 0.88	4 / 0.13	5 / 0.17	24 / 0.96	17 / 0.67	23 / 0.92	25 / 1	4 / 0.13	12 / 0.46	8 / 0.3	3 / 0.09	3 / 0.09	3 / 0.09	7 / 0.25	5 / 0.17
p 14	8 / 1	2 / 0.15	2 / 0.15	7 / 0.86	3 / 0.29	7 / 0.86	3 / 0.29	8 / 1	6 / 0.72	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	2 / 0.15
p 15	7 / 0.67	6 / 0.56	6 / 0.56	9 / 0.89	9 / 0.89	9 / 0.89	9 / 0.89	6 / 0.56	6 / 0.56	4 / 0.34	4 / 0.34	4 / 0.34	4 / 0.34	4 / 0.34	4 / 0.34
p 16	5 / 0.31	3 / 0.16	1 / 0	14 / 1	6 / 0.39	5 / 0.31	11 / 0.77	3 / 0.16	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 17	3 / 0.5	1 / 0	3 / 0.5	5 / 1	5 / 1	5 / 1	5 / 1	3 / 0.5	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0
p 18	4 / 0.5	4 / 0.5	4 / 0.5	6 / 0.84	6 / 0.84	6 / 0.84	6 / 0.84	4 / 0.5	4 / 0.5	3 / 0.34	3 / 0.34	3 / 0.34	6 / 0.84	6 / 0.84	6 / 0.84
p 19	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	2 / 0.25	1 / 0	2 / 0.25
p 20	7 / 0.29	5 / 0.2	4 / 0.15	16 / 0.72	19 / 0.86	15 / 0.67	17 / 0.77	2 / 0.05	3 / 0.1	1 / 0	1 / 0	2 / 0.05	2 / 0.05	1 / 0	3 / 0.1
p 21	10 / 1	5 / 0.45	5 / 0.45	9 / 0.89	9 / 0.89	9 / 0.89	9 / 0.89	5 / 0.45	5 / 0.45	4 / 0.34	4 / 0.34	4 / 0.34	4 / 0.34	4 / 0.34	4 / 0.34
p 23	7 / 0.86	1 / 0	1 / 0	7 / 0.86	7 / 0.86	7 / 0.86	7 / 0.86	1 / 0	7 / 0.86	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0	1 / 0

Topologically Consistent Selective Progressive Transmission

Padraig Corcoran, Peter Mooney

Department of Computer Science, National University of Ireland Maynooth, Ireland.
{padraigc, peter.mooney}@cs.nuim.ie

Abstract. Progressive transmission represents a viable solution to the challenges presented by the transmission of large vector data sets over the Internet. Previous implementations have considered progressive transmission as the reverse of map generalization. In an adaptive or selective progressive transmission strategy, the order of transmission can vary between clients and generally will not equal the reverse of the corresponding generalization. In this context, we propose that generalization can only represent a pre-processing step to a distinct selective progressive transmission process. One of the greatest challenges in implementation of such an approach is determining topological equivalence with the original map. We propose this problem may be represented in the form of three challenges. We perform a formal mathematical analysis of solutions to these challenges and present a corresponding implementation.

1 Introduction

The delivery of spatial data over the Internet, known as Web-GIS, is quickly becoming the most popular medium for obtaining such data (Bertolotto 2007). Due to limitations in network bandwidth there is a trade-off between the requirements to deliver data of high detail and to deliver it within reasonable time. To tackle these conflicting requirements, many researchers have considered a progressive transmission strategy (Bertolotto 2007). Many existing approaches to progressive transmission function as

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_26,

follows. A sequence of generalization levels of the map in question are pre-computed where each level contains less detail than the previous (Hamid et al. 2010; Zhang et al. 2010; Yang et al. 2007). To perform progressive transmission the levels are progressively sent to the user in reverse order to that of the generalization. Progressive transmission is achieved through a process of refinement which either sends the entire level at each step or only the additions required to compute the current level from the previous. In essence, this process of refinement becomes the inverse of the generalization process (Ai et al. 2005). The goal of any generalisation process is to produce a result which achieves a set of objectives (Jones and Ware 2005). Much geographical analysis is a function of map topology; therefore, one important objective of generalisation is that all resulting maps have equivalent topology to the original map; that is, all simplifications are topologically consistent (Weibel 1996). We refer to the map of least detail resulting from generalization as the base map. Ideally, the client should be able to perform analysis using data at any level of refinement and terminate the transmission when the data reaches a desired level of detail (Ai et al. 2005). By implication of the fact that progressive transmission is the inverse of generalization, all levels of the transmission will also satisfy the same objectives as the corresponding generalizations. For example, if all generalization levels are topologically equivalent to the original map this will also be the case for all levels in the progressive transmission.

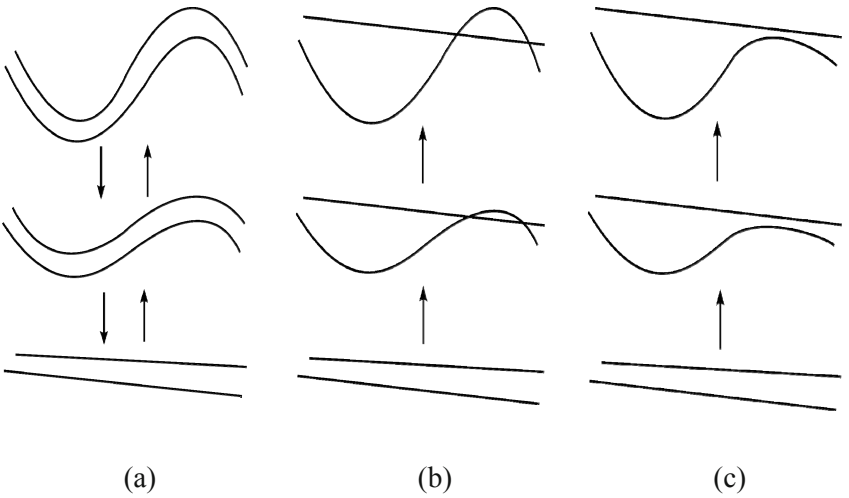


Fig. 1. The arrows pointing down and up represent generalization and progressive transmission processes respectively.

Consider the original map represented in the top diagram of Figure 1(a) which contains two line features. A generalization process, represented by arrows pointing down, consisting of two steps is applied to these features. This results in a topologically consistent base map which contains two straight line features and is represented in the bottom diagram of Figure 1(a). This base map is initially sent to all clients. Next, using the refinement process, details removed through the generalization process are progressively sent and integrated; this process is represented by arrows pointing up in Figure 1(a). This traditional approach to progressive transmission is not adaptive to varying client requirements. All clients receive the same initial base map and the same order of feature refinement. We propose that, in order for progressive transmission to be adaptive to such requirements, map generalization and refinement must be considered as two distinct processes. Successful implementation of each of these presents unique challenges. Firstly, the map must be first generalized subject to user requirements. The problem of adaptive generalization has previously been considered. Kulik et al. (2005) proposed a map generalization technique where features deemed more important by the user are represented at a greater level of detail. For example, a walker may desire paths to be represented with high detail and roads with low detail. Secondly the refinement of the base map must also be adaptive and not simply considered the reverse of the earlier generalization. A user may require a unique refinement process where, for example, only specific features deemed important are refined. Also, these requirements may change in real time during transmission. An adaptive refinement process would ideally accommodate such user requirements.

In this paper, we focus on the second of the challenges presented above. We present a methodology which allows adaptive refinement of map features. In our design, all clients receive the same base map. These are then iteratively refined in a selective manner such that features deemed important are refined while features deemed unimportant remain constant or receive little refinement. Feature importance can be determined based on client requirements. For example, the user could be asked to select important features from a list containing roads, trails, parks, etc. This concept of a selective progressive transmission strategy was mentioned briefly by Bertolotto and Egenhofer (2001). In the context of such a transmission strategy, ensuring topological equivalence between refinements of the base and the original map presents a novel challenge. Consider the progressive transmission of the map displayed in the bottom diagram of Figure 1(b). In this example the lower line feature is refined while the upper line feature remains constant. This would be the case if the user only required the lower feature to be represented at a high level of detail. Simply applying

the reverse of the generalization process to only the lower feature introduces a topological inconsistency. That is an intersection between line features is introduced; this is evident in the middle and top diagrams of Figure 1(b). To overcome this issue we propose a novel topological consistent refinement strategy where only features deemed important are refined but this refinement is constrained such that no topological inconsistencies are introduced. This concept is illustrated in Figure 1(c).

The layout of this paper is as follows. In section 2, we introduce the generalization process used to generate the base map which is initially sent to each client. Section 3 describes the proposed selective progressive transmission methodology. Sections 4 and 5 present results and draw conclusions, respectively.

2 Map Generalization

Jones (1997) describes eight types of generalization operators. These are elimination, simplification, typification, exaggeration, enhancement, collapse, amalgamation, and displacement. Simplification methods, which represent the focus of this work, attempt to generalize features by reducing the number of vertices used for representation. As introduced briefly in Section 1, the goal of any generalisation process is to produce a result which achieves a set of objectives (Jones and Ware 2005). Weibel (1996) identified four classes of objectives which such a process may aim to satisfy. These are shape (Gestalt), semantic, metric, and topological objectives. Shape objectives require that successive levels of generalization represent an intuitive shape evolution (Latecki and Lakmper 1999). Semantic objectives integrate information about a feature's semantics when deriving generalization. For example, a line feature may be generalized differently if it represents a road as opposed to a river. Metric objectives aim to achieve the best possible result in terms of an error criterion. For example, this could be the result which minimizes the overall deviation from the original map. Finally, topological objectives are primarily concerned with the need to ensure that the simplified representations retain the original relationships of containment and connectivity (Jones and Ware 2005). Two maps with equal topology are said to be topologically equivalent (Kuijpers et al. 1995; Cicerone et al. 2002).

In this paper, we implement a generalization strategy which satisfies both shape and topological objectives. In order to achieve this, two components are necessary. Firstly, to satisfy shape objectives, a function capable of determining relative vertex significance is required such that verti-

ces with least significance are removed first. Secondly, to satisfy topological objectives, a method which takes as input a map and a corresponding simplification determines if both are topologically equivalent. At each simplification step the least significant vertex, such that the corresponding map is topologically consistent, is removed. To determine vertex significance the method of Latecki and Lakmper (1999) was used. Using this approach, individual vertex significance is a function of adjacent line segment lengths and corresponding turning angle. Determining topological equivalence between a map and corresponding simplification has been the focus of much previous research (de Berg et al. 1998; Saalfeld 1999; da Silva and Wu 2006). Existing techniques for determining the topological consistency of a simplification attempt to provide a solution to the following challenge.

Challenge 1 – Simplification Topological Equivalence: *Given two topological spaces X and Y and two pairs of objects (Ax, Bx) and (Ay, By) in X and Y respectively, such that By equals Bx and Ay is a simplification of Ax; determine if the topological relation which exists between the pair (Ax, Bx) is equivalent to that which exist between the pair (Ay, By).*

This challenge is illustrated using Figure 2 where the task is to determine if the topological relationship between *(Ax, Bx)* equals that between *(Ay, By)*. *Ay* is a simplification of A_X obtained by removing the vertex v_i while B_Y is equal to B_X.

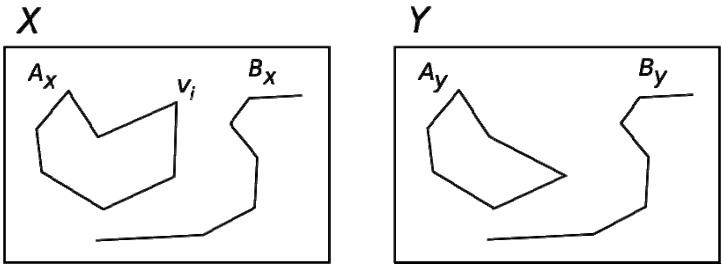
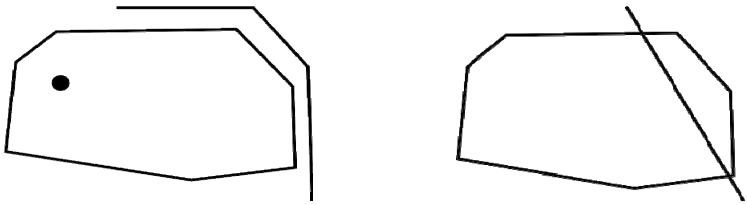


Fig. 2. A graphical representation of the Simplification Topological Equivalence challenge. *X* corresponds to the original map while *Y* corresponds to a simplified map.

All topological relationships between map features can be classified as planar or non-planar (Corcoran et al., 2010). Consider the simple map in Figure 3(a) which contains a polygon, a line, and a point feature. No lines or edges in this map cross without forming a vertex; therefore, all such topological relationships are referred to as planar. Next, consider the sim-

ple map in Figure 3(b) which contains a polygon and line feature. The line crosses the polygon without forming a vertex; we, therefore, refer to such a topological relationship as non-planar.



(a) (b)

Fig. 3. Planar and non-planar topologies are shown in (a) and (b) respectively.

To determine if a simplification is topologically consistent with respect to a planar relationship between a point and a line, the strategy of Saalfeld (1999) is used. Saalfeld (1999) proved a simplification is topologically consistent with respect to such a relationship if the point in question does not lie inside a region between the original and simplified contours. Such regions are referred to as bounded faces (Saalfeld 1999) and are defined as follows. Let *I(p, X)* be a function which returns an integer representing the number of times a half-ray from a point *p* in any fixed direction intersects a contour *X*. Let *C* be a simple contour and *C′* its corresponding simplification. If *(I(p, C) + I(p, C′))mod 2 = 1, p* lies inside a bounded face formed by *C* and *C′*; otherwise if *(I(p, C) + I(p, C′)) mod 2 = 0*, *p* lies outside a bounded face formed by *C* and *C′*. For example, consider the contour *C* and its corresponding simplification *C′* in Figure 4(a). This simplification is topologically inconsistent with respect to the point *p* because this point changes sidedness and correspondingly lies in a bounded face. To determine topological consistency with respect to a planar relationship between two lines which do not intersect, the strategy of da Silva and Wu (2006) is used. Da Silva and Wu (2006) proved a simplification is topologically consistent with respect to such a relationship if no segment endpoint falls inside a bounded face formed by each segment in the simplification and its corresponding original contour. For example consider the contour *C* and its corresponding simplification *C′* in Figure 4(b). *C′* is topologically inconsistent with respect to the line segment l_1l_2 because the endpoint l_1 lines in a bounded face formed by the segment v_3v_5 in *C′* and its corresponding contour in *C*.

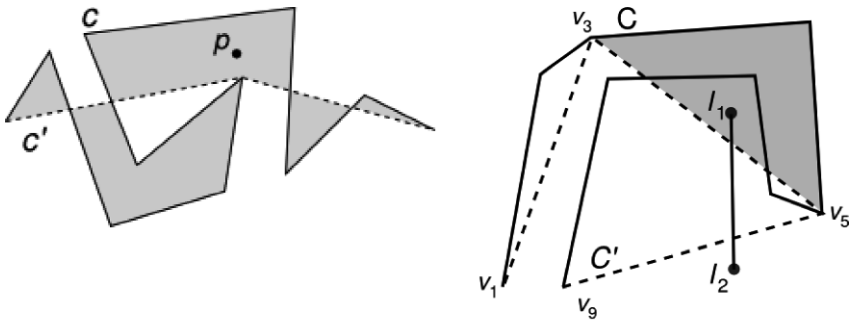


Fig. 4. In (a) and (b) the contour C is simplified to form the contour C'. In (a) the bounded faces which exist between both complete contours are coloured grey. In (b) the bounded face which exists between a single line segment in C' and its corresponding contour in C is coloured grey.

We now provide a revision of the original proof presented by da Silva and Wu (2006) which we will later draw upon in Section 3. This proof contains a lemma and theorem. In the lemma, the case where the simplification contains a single line segment is proven. The theorem generalizes this to the case where the simplification contains one or more line segments. Let C be a simple contour and C' a corresponding simple simplification which is a single line segment. Let be $l_i l_j$ a line segment which does not intersect C. A scene containing such geometry is illustrated in Figure 5.

Lemma 1. C' is a consistent simplification of C with respect to $l_i l_j$ if, and only if, l_i and l_j do not lie in a bounded face formed by C and C'.

Proof. There exists a contour B that completes both C and C' to simple polygons, denoted BC and BC' respectively, such that BC contains l_i and l_j and $l_i l_j$ does not intersect BC. Such a contour is shown in Figure 5. Consider the case where $l_i l_j$ intersects C' as illustrated in Figure 5. $l_i l_j$ does not intersect B and line segments which do not overlap can only intersect in a single point. Therefore, if $l_i l_j$ intersects BC', it can only do so in a single point lying along C'. By the point-in-polygon criterion, a segment which intersects a polygon a single time will have one endpoint lying inside and the other endpoint lying outside the polygon. Consequently, if such an intersection occurs one endpoint of $l_i l_j$ will lie outside BC' in a bounded face between C and C' and be determined inconsistent. On the other hand, if $l_i l_j$ does not intersect C' no endpoint of $l_i l_j$ will lie outside BC' in a bounded

face between C and C'. The simplification will then be determined consistent.

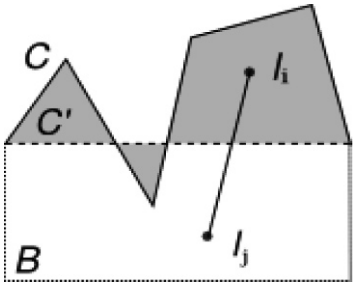


Fig. 5. C' represents an inconsistent simplification with respect to the line segment $l_i l_j$.

Theorem 1. The contour C', which contains one or more line segments, is a consistent simplification of C with respect to $l_i l_j$ if, and only if, each line segment in C' is determined consistent with respect to $l_i l_j$ by Lemma 1.

Proof. If each individual line segment in C' is determined consistent with respect to $l_i l_j$ by Lemma 1, then $l_i l_j$ does not intersect C'. If $l_i l_j$ intersects one or more line segments in C', this will be determined by the corresponding evaluation of Lemma 1.

To ensure that all simplifications are topologically consistent with respect to non-planar topology, we must ensure that no existing line intersections are removed and no new intersections are introduced. To ensure no line intersections are removed, the strategy of Kulik et al. (2005) and Weihua (2008) was employed. In this strategy, line intersections are maintained by marking all line segments that contain intersections as *unremovable*. To insure no new line intersections are introduced, the strategy of da Silva and Wu (2006) presented above was used.

3 Selective Progressive Transmission

In this section, we describe a selective progressive transmission strategy which satisfies shape and topological objectives. The novelty of our approach is that the decision(s) regarding which features need refinement can be decided on the fly. In addition to this feature, the order of refinement is not necessarily the inverse of the initial generalization. Topological objectives require that all refinements are topologically equivalent to the base

map or previous refinement and, in turn, the corresponding original map. That is, all refinements are topologically consistent. In order to implement such a transmission strategy, two components are necessary. Firstly to satisfy shape objectives, a function capable of determining relative vertex significance is required such that vertices with the greatest significance are added first. Secondly, to satisfy topological objectives, a method which can determine if a given refinement is topologically consistent is required. At each refinement step, the most significant vertex from the features which require refinement, such that the corresponding map is topologically consistent, is added. To determine vertex significance, the method of Latecki and Lakmper (1999), introduced in Section 2, is used. Determining if a given refinement is topologically consistent cannot always be achieved by applying existing methods, such as those presented in Section 2, which compute the topological consistency of simplifications. In certain cases, new techniques must be developed. In this section, we identify cases where existing techniques can be used and developing new techniques where needed. The problem of determining topological consistency of a refinement can be posed in two different ways. The first challenge is to determine topological equivalence between the less detailed or simplified map in question and its corresponding refinement and is presented formally as Challenge 2.

Challenge 2 – Refinement Topological Equivalence by Comparison to the Simplified Map*: Given two topological spaces X and Y and two pairs of objects (Ax, Bx) and (Ay, By) in X and Y respectively, such that Bx equals By and Ay is a refinement of Ax; determine if the topological relation which exists between the pair (Ax, Bx) is equivalent to the topological relation which exist between the pair (Ay, By).*

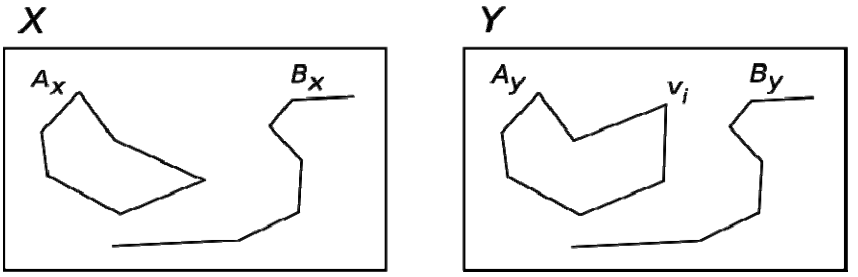


Fig. 6. Graphical representation of determining Refinement Topological Equivalence by Comparison to Simplified Map. *X* represents the simplified map and *Y* a corresponding refinement.

This challenge is illustrated using Figure 6 where the task requires us to determine if the topological relationship between the pair *(Ax, Bx)* equals that between the pair *(Ay, By)*. *Ay* is a refinement of A_X obtained by adding the vertex v_i while *By* is equal to B_X.

In the second approach to determining topological consistency one attempts to determine topological equivalence between the original map in question and the corresponding refinement. The refinement of a simplified map is in fact a simplification of the original map. Therefore, if topological equivalence can be determined, this implies the refinement is topologically consistent. This is introduced formally as Challenge 3.

Challenge 3 – Refinement of Topological Equivalence by Comparison to the Original Map*: Given two topological spaces X and Y and two pairs of objects (Ax, Bx) and (Ay, By) in X and Y, respectively, such that Ay is a simplification of Ax and By is a simplification of Bx, determine if the topological relation which exists between the pair (Ax, Bx) is equivalent to the topological relation which exists between the pair (Ay, By).*

This is illustrated using Figure 7 where the task requires us to determine if the topological relationship between the pair *(Ax, Bx)* equals that between the pair *(Ay, By)*. *Ay* is a simplification of *Ax* while *By* is a simplification of B_X.

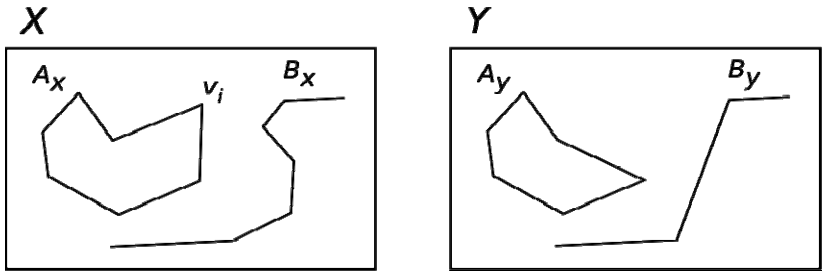


Fig. 7. Graphical representation of determining Refinement of Topological Equivalence by Comparison to Simplified Map. *X* corresponds to the original map while *Y* corresponds to a refinement of a simplified map.

Although similar, there is one major difference between Challenge 3 and Challenge 1 presented in Section 2. In Challenge 1, when determining topological equivalence, it is assumed that only a single object is simplified while all others remain constant. Referring back to the illustration of Challenge 1 in Figure 2, we see that A_X is simplified to form *Ay* while B_X is

equal to *By*. In Challenge 3, when determining topological equivalence, both objects in question may be simplified versions of their original forms. This is illustrated in Figure 7. Using Challenges 2 and 3 as different approaches to posing the problem, the following two sections describe how planar and non-planar topological consistency of a refinement may be determined.

3.1 Planar Topological Equivalence

Using the approach of Saalfeld (1999) presented in Section 2, the topological consistency of a refinement with respect to a relationship between a point and a line may be determined through implementation of solutions to Challenges 2 and 3, that is, by comparison to the simplified map or the original map. We present a solution which adopts the former of these approaches. Let C and C' be a simple contour and a corresponding simple refinement, respectively; let p be a point feature. A scene containing such geometry is illustrated in Figure 8.

Theorem 2. C' is a consistent refinement of C with respect to p if, and only if, p does not lie in a bounded face formed by C and C'.

Proof. By viewing C as a simplification of C', topological consistency with respect to p can be determined using the strategy of Saalfeld (1999). This is due to the fact that the proof of this method is not dependent on which of the two contours in question is a simplification of the other.

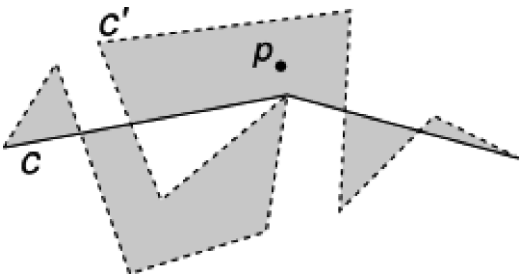


Fig. 8. The contour C is refined to form the contour C'. The bounded faces which exist between both contours are coloured grey.

For a contour containing n line segments and a single point, Theorem 2 can be evaluated in $O(n)$ time complexity. Theorem 2 proved that it is possible to determine the topological consistency of a refinement with respect

to a planar relationship with a point by restating the problem as one of simplification and using the approach of Saalfeld (1999). The second planar relationship we must consider when determining topological consistency of a refinement is the relationship of non-intersecting lines. In section 2, we proved that the topological consistency of a simplification with respect to this relationship may be determined using the strategy of da Silva and Wu (2006). Unfortunately, this method cannot be used to determine the topological consistency of a refinement by restating the problem as one of simplification like previously. We now demonstrate why this is the case.

Firstly, we analyse the approach of Challenge 2 where one attempts to determine topological consistency by comparison to the simplified map. Consider the contour C and its corresponding refinement C' in Figure 9. C' is a single segment v_1v_2 which is refined by replacing it with the two segments v_1v_3 and v_3v_2. We wish to determine if C' is a topologically consistent refinement with respect to the planar relationship which exists between C and the segment l_1l_2. Using the method of da Silva and Wu (2006), we consider C to be a simplification of C'. Applying Theorem 1 in this context states that C is a consistent simplification of C' with respect to l_1l_2 if, and only if, each line segment in C is determined consistent with respect to l_1l_2 by Lemma 1. C contains a single line segment and applying Lemma 1 to this states that C is a consistent simplification of C' with respect to l_1l_2 if, and only if, l_1 and l_2 do not lie in a bounded face formed by C' and C. A single bounded face exists between C' and C and this is represented by the colour grey in Figure 9. No endpoint of l_1l_2 lies in this region and this refinement is determined consistent. Clearly this is not the case because an intersection with l_1l_2 has been introduced with C'. Therefore, applying the method of da Silva and Wu (2006) in this manner cannot determine if a refinement is consistent with respect to this topological relationship.

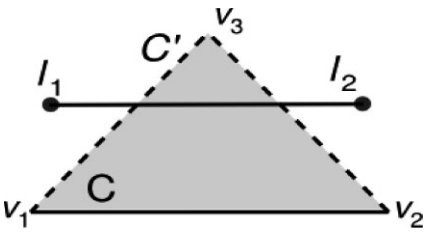


Fig. 9. The contour C is refined to form the contour C'. The bounded face which exists between both contours is coloured grey.

Next, we analyze the approach where one attempts to determine topological equivalence by comparison to the original map; that is Challenge 3. Consider the contour *C* which is a simplification of the original contour *CO* and is refined to form *C′* in Figure 10. We wish to determine the topological consistency of the refinement *C′* with respect to the contour *D* which is a simplification of *DO*. Representing *C′* as a simplification of *CO* and applying the method of da Silva and Wu (2006) determines *C′* to be topologically consistent. This is because no endpoint of D lies in a bounded face, represented by the colour grey in Figure 10, formed by *CO* and *C′*. Clearly this is not the case because the refinement introduces an intersection with *D*. As before, applying the method of da Silva and Wu (2006) in this manner cannot determine if a refinement is consistent with respect to this topological relationship.

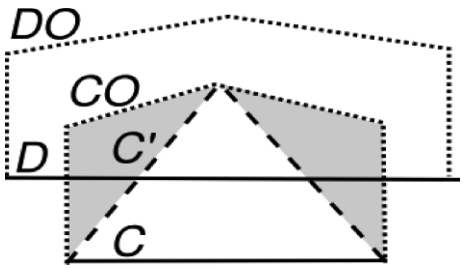


Fig. 10. The contour *C* (solid line) is a simplification of *CO* (dotted line) and is refined to form the contour *C′* (dashed line). The bounded faces which exist between both contours are coloured grey.

To understand the reasons why the method of da Silva and Wu (2006) fails to correctly determine topological equivalence of refinements in both Challenges 2 and 3, we refer back to the proof of da Silva and Wu (2006) in Theorem 1. This proof is based on the assumption that a set of bounded faces between each segment in the simplified or reduced contour and its corresponding original or detailed contour can be constructed such that each boundary contains only a single segment which the segment $l_i l_j$ can intersect. In the context of refinement this assumption is not valid. Each of the bounded faces between the segments in the reduced contour and the corresponding refined contour contain more than a single segment which $l_i l_j$ can intersect. For example, the bounded face in Figure 9 contains two segments which $l_1 l_2$ can and does intersect. This allows the segment to enter and leave the bounded face in question. Consequently, no endpoint lies in the face. This situation is also present in Figure 10 where *D* intersects the boundary of each bounded face twice. This may not only result in con-

tours which intersect each other but also contours which self-intersect. To overcome these issues we propose a different solution to Challenge 2. We examine if any line segment in one contour intersect any line segment in the other. If no intersection is found, the planar topological relationship between the contours in question is consistent. In reference to Figure 9 this would involve determining if the following pairs of line segments intersect: firstly l_1l_2 and v_1v_3 and secondly l_1l_2 and v_3v_2. Since intersections occur, this refinement would be correctly determined inconsistent. For two contours containing n and m line segments, respectively, this proposed solution requires $O(nm)$ time complexity. To remove the requirement to evaluate all possible intersections between all pairs of contours in the map, the following strategy is used. Saalfeld (1999) proved that two simplified contours may only intersect if the convex hulls of their corresponding original contours intersect. Using this fact we precompute the pairs of contours that may possibly intersect and only evaluate intersections between such pairs at all refinement steps. This is similar to the safe-set approach of Mantler and Snoeyink (2000).

3.2 Non-Planar Topological Equivalence

To ensure that all refinements are topologically consistent with respect to non-planar topology, we must ensure that no existing line intersections are removed and no new intersections are introduced. To ensure no line intersections are removed, the strategy Kulik et al. (2005) and Weihua (2008), which we introduced in Section 2, is used. Line intersections are maintained by marking all line segments that contain intersections as *unremovable* by the refinement process.

4 Results

In this section, we demonstrate the effectiveness of the proposed methodology in generating a selective progressive transmission which satisfies both topological and shape objectives. Simplification and the selective progressive transmission algorithms were implemented in the C++ programming language. The point, line, and polygon data structures from the Computational Geometry Algorithms Library (CGAL) (Giezeman and Wesselink 2008) were used to represent all features. Analysis was performed using a data set extracted from OpenStreetMap (OSM) which is displayed in Figure 11.

Fig. 11. A sample OpenStreetMap data set is shown. This map corresponds to row two of Table 1.

This data set contains three polygons having a total of 486 vertices and 52 lines having a total of 1270 vertices; this information is represented in the second row of Table 1. A base map for this data set was computed using the simplification methodology of Section 2 and is displayed in Figure 12. The base map contains 364 line and 64 polygon vertices; the third row of Table 1 contains this information. This corresponds to a 76% reduction in data size. It is evident that all features in the base map have been simplified and this map is topologically equivalent to the original map. A selective progressive transmission strategy was applied to this base map where polygons are deemed important and refined while lines are deemed unimportant and remain constant. This is one of many possible refinement strategies which could have been applied and the exact strategy is ultimately determined by user requirements. For example, the user could be asked to select important features from a list containing roads, trails, parks, etc. The final result of this refinement process is shown in Figure 13 where it is evident that all polygons are represented with high detail and all lines are represented with low detail. It is also evident that the result is topologically equivalent to the original map in Figure 11. This refinement result contains 364 line vertices and 478 polygon vertices; the sixth row of Table 1 contains the information.

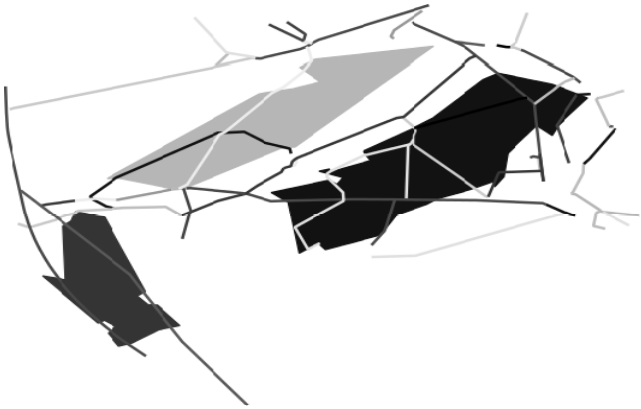


Fig. 12. The map in Figure 11 is simplified to form a base map. This map corresponds to row three of Table 1.

Fig. 13. The map in Figure 12 is refined using a selective progressive transmission strategy. This map corresponds to row six of Table 1.

To highlight the topological preserving property of the proposed selective progressive transmission methodology, we visualize a small spatial area of the original map in Figure 14(a). The corresponding base map is shown in Figure 14(b). Using the same strategy as before where only polygons are refined, two intermediate steps and the final result of refinement are shown in Figure 14(c), Figure 14(d) and Figure 14(e), respectively. All refinements are topologically consistent. The number of polygon and line vertices at each step is represented in rows four, five, and six of Table 1. Figure 14(f) shows the result of the same refinement strategy but where no effort is made to maintain topological consistency. That is, the polygons in

the original map are equal to these polygons in the resulting refinement. It is evident that two intersections have been introduced between the upper polygon and the line in the centre of the figure. Consequently the refinement is not topologically consistent. The final row of Table 1 shows the number of polygon and line vertices in this topologically inconsistent result. Only a relatively small number of extra polygon vertices are present in this result compared to the topologically consistent result (comparison of rows 6 and 7 in Table 1). The proposed method correctly determined not to add these extra vertices in order to maintain topology equivalence to the original map.

The benefit of our proposed selective progressive transmission strategy is evident from closer analysis of Table 1. Consider the case where the two requirements of the user are to have polygons represented in high detail and to have a topologically consistent map. Using existing progressive transmission strategies the entire data set would be transmitted in order to ensure this. In the case of this data set, that is 1756 vertices. Using the proposed methodology, both goals can be achieved by only transmitting 842 vertices (Table 1, row 6) (only 47% of the original data set size). The same analysis was performed on five other data sets with each containing over 1500 vertices. In all cases, we found the reduction in data size requiring transmission was over 50%. Running on an Intel 2.8 GHz dual core processor simplification of the data set in Figure 11 was achieved in less than 3 seconds (producing Figure 12 from Figure 11). Complete refinement of this base map was achieved in less than 2 seconds (producing Figure 13 from Figure 12).

Table 1. The numbers of polygon and line vertices for each stage of the selective progressive transmission strategy are shown.

	No. Line Vertex	No. Polygon Vertex	Total No. Vertex
Original Map	1270	486	1756
Base Map	364	64	428
Refinement Stage 1 – Topologically Consistent	364	139	503
Refinement Stage 2 - Topologically Consistent	364	289	653
Refinement Result - Topologically Consistent	364	478	842
Refinement Result - Not Topologically Consistent	364	486	850

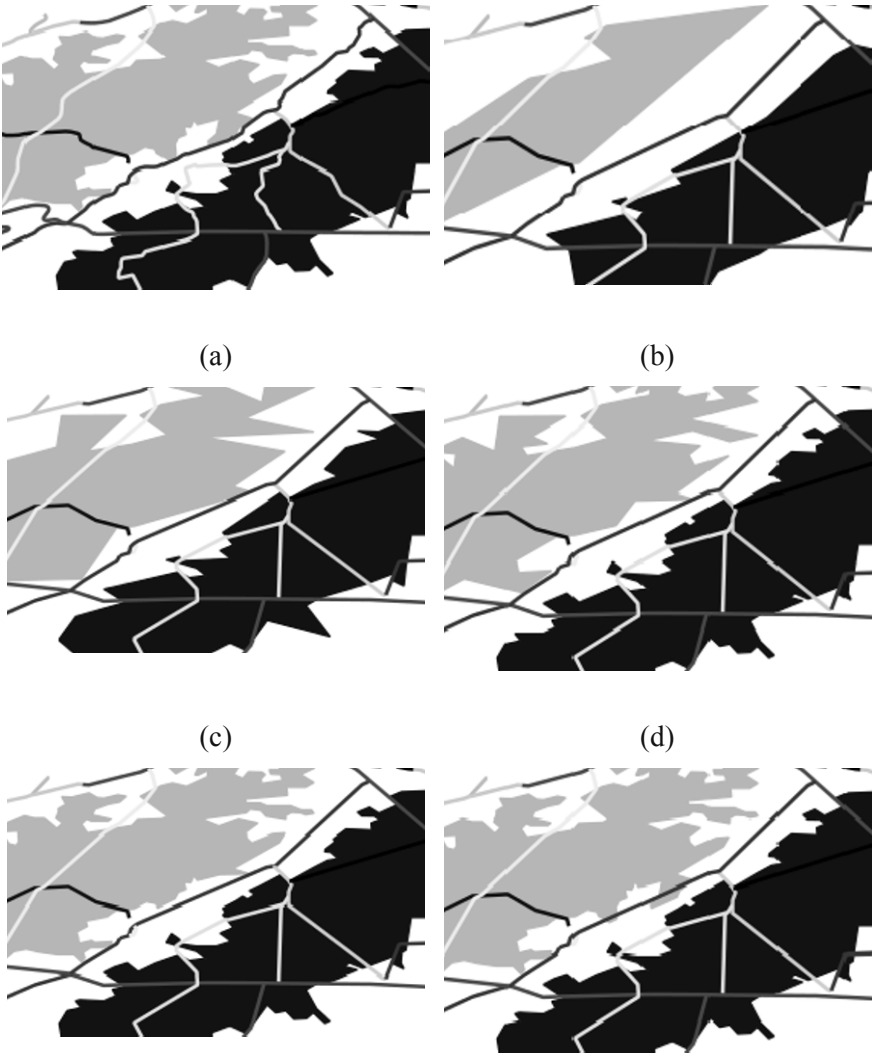


Fig. 14. A closer analysis of the proposed selective progressive transmission strategy applied to the data set of Figure 11 is shown.

5 Conclusions

This paper presents a novel model of selective progressive transmission for vector data which is adaptive to user requirements. The model differs from existing implementations which view progressive transmission as the inverse of generalization. In this paper, we proposed that in order for generalization and progressive transmission to be adaptive and satisfy user requirements, they must be viewed as two distinct processes. To demonstrate this, a selective progressive transmission strategy which satisfies shape and topological objectives is presented. Determining if a given refinement is topologically consistent represents the greatest challenge in implementation of such a system. To achieve this, an in-depth mathematical analysis and corresponding solution are presented. Results on a real data set show the proposed methodology can satisfy these requirements while reducing the data set size which must be transmitted.

In the current implementation of this work, simplification and refinement are performed on a single computer. In future work, we intend to implement the proposed methodology using a client-server model. In such a model, simplification and refinement would be performed on the server side. The base map would then be transmitted to the client followed by map refinements which would be integrated by the client device.

References

Ai, T., Li, Z., Liu, Y. (2005) Developments in Spatial Data Handling. In: Progressive Transmission of Vector Data Based on Changes Accumulation Model, Springer Berlin Heidelberg, pp. 85–96.

Bertolotto, M. (2007) Progressive Techniques for Efficient Vector Map Data Transmission. In: Spatial Data on the Web: Modelling and Management. Springer-Verlag. pp. 65–84.

Bertolotto, M., Egenhofer, M. J. (2001) Progressive Transmission of Vector Map Data Over the World Wide Web. In: Geoinformatica 5(4), pp. 345–373.

Cicerone, S., Frigioni, D., Felice, P. D. (2002) A General Strategy for Decomposing Topological Invariants of Spatial Databases and an Application. In: Data and Knowledge Engineering 42 (1), pp. 57–87.

Corcoran, P. and Mooney, P. and and Winstanley A.C. (2011) Planar and Non-Planar Topologically Consistent Vector Map Simplification. In: International Journal of Geographical Information Science, In Press.

da Silva, A. C. G., Wu, S.-T. (2006) A Robust Strategy for Handling Linear Features in Topologically Consistent Polyline Simplification. In: GeoInfo. pp. 19–34.

de Berg, M., van Kreveld, M., Schirra, S. (1998) Topologically Correct Subdivision Simplification Using the Bandwidth Criterion. In: Cartography and Geographic Information Systems 25(4), pp. 243–257.

Giezeman, G.J. and Wesselink, W. (2008) 2D Polygons. In: C.E. Board, ed. CGAL User and Reference Manual.

Hamid, A. A., Ahmed, M., Helmy, Y. (2010) Enhanced Progressive Vector Data Transmission For Mobile Geographic Information Systems (MGIS). In: Innovations and Advances in Computer Sciences and Engineering, Springer Netherlands, pp. 61–66.

Jones, C. (1997) Geographical Information Systems and Computer Cartography. Prentice Hall.

Jones, C., Ware, J. (2005) Map generalization in the Web age. International Journal of Geographical Information Science 19(8-9), 859 – 870.

Kuijpers, B., Paredaens, J., Bussche, J. V. d. (1995) Lossless Representation of Topological Spatial Data. In: SSD '95: Proceedings of the 4th International-Symposium on Advances in Spatial Databases. Springer-Verlag, London, UK, pp. 1–13.

Kulik, L., Duckham, M., Egenhofer, M. (2005) Ontology-Driven Map Generalization. Journal of Visual Languages and Computing 16 (3), 245–267.

Latecki, L., Lakmper, R. (1999) Convexity rule for shape decomposition based on discrete contour evolution. Computer Vision and Image Understanding 73 (3), 441 – 454.

Mantler, A. and Snoeyink, J. (2000) Safe sets for line simplification, In: Proceedings of the 10th Annual Fall Workshop on Computational Geometry. Stony Brook, New York.

Saalfeld, A. (1999) Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm. Cartography and Geographic Information Science 26 (1), 7–18.

Weibel, R. (1996) Advances in GIS Research II (Proceedings 7th International Symposium on Spatial Data Handling). London: Taylor & Francis, Ch. A Typology of Constraints to Line Simplification, pp. 533–546.

Weihua, D., (2008) Generating On-Demand Web Mapping Through Progressive Generalization. In: International Workshop on Education Technology and Training, 2 pp. 163–166.

Yang, B., Purves, R., Weibel, R. (2007) Efficient Transmission of Vector Data Over the Internet. International Journal of Geographical Information Science 21(2), pp. 215–237.

Zhang, L., Kang, Z., Li, J., Yang, L. (2010) Web-Based Terrain and Vector Maps Visualization for Wenchuan Earthquake. International Journal of Applied Earth Observation and Geoinformation 12(6), pp. 439 – 447.

Erratum to:

Advancing Geoinformation Science for a Changing World

S.C.M. Geertman, W.P. Reinhardt, F.J. Toppen

S.C.M. Geertman et al. (eds.), *Advancing Geoinformation Science for a Changing World*,
Lecture Notes in Geoinformation and Cartography 1, DOI 10.1007/978-3-642-19789-5_13,

10.1007/978-3-642-19789-5_27

In Chapter 13, there is a spelling mistake in the surname of the third author. In particular "Linda Seel" replace by "Linda See"

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/978-3-642-19789-5_13