

# 11th International Conference on Geographic Information Science

GIScience 2021, September 27–30, 2021, Poznań, Poland  
(Virtual Conference)

Part II

Edited by

Krzysztof Janowicz

Judith A. Verstegen



*Editors*

**Krzysztof Janowicz**

University of California, Santa Barbara, USA  
janowicz@ucsb.edu

**Judith A. Verstegen** 

Wageningen University, Wageningen, The Netherlands  
judith.verstegen@wur.nl

*ACM Classification 2012*

Information systems → Geographic information systems; Human-centered computing → Human computer interaction (HCI); Human-centered computing → Visualization; Theory of computation → Computational geometry; Computing methodologies → Machine learning; Information systems → Spatial-temporal systems

**ISBN 978-3-95977-208-2**

*Published online and open access by*

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-208-2>.

*Publication date*

September, 2021

*Bibliographic information published by the Deutsche Nationalbibliothek*

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

*License*

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.GIScience.2021.II.0

ISBN 978-3-95977-208-2

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

## LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

### *Editorial Board*

- Luca Aceto (*Chair*, Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Mikolaj Bojanczyk (University of Warsaw, PL)
- Roberto Di Cosmo (Inria and Université de Paris, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University - Brno, CZ)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (University of Oxford, GB)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)

**ISSN 1868-8969**

**<https://www.dagstuhl.de/lipics>**



## ■ Contents

Preface	
<i>Krzysztof Janowicz and Judith A. Versteegen</i> .....	0:vii
Conference Organization	
.....	0:ix
List of Authors	
.....	0:xiii

### Regular Papers

Adaptive Voronoi Masking: A Method to Protect Confidential Discrete Spatial Data	
<i>Fiona Polzin and Ourania Kounadi</i> .....	1:1–1:17
Reproducible Research and GIScience: An Evaluation Using GIScience Conference Papers	
<i>Frank O. Ostermann, Daniel Nüst, Carlos Granell, Barbara Hofer, and Markus Konkol</i> .....	2:1–2:16
Comparison of Simulated Fast and Green Routes for Cyclists and Pedestrians	
<i>Christina Ludwig, Sven Lautenbach, Eva-Marie Schömann, and Alexander Zipf</i> ...	3:1–3:15
A Clustering-Based Framework for Individual Travel Behaviour Change Detection	
<i>Ye Hong, Yanan Xin, Henry Martin, Dominik Bucher, and Martin Raubal</i> .....	4:1–4:15
Will You Take This Turn? Gaze-Based Turning Activity Recognition During Navigation	
<i>Negar Alinaghi, Markus Kattenbeck, Antonia Golab, and Ioannis Giannopoulos</i> ...	5:1–5:16
Bicriteria Aggregation of Polygons via Graph Cuts	
<i>Peter Rottmann, Anne Driemel, Herman Haverkort, Heiko Röglin, and Jan-Henrik Haurert</i> .....	6:1–6:16
Coordinated Schematization for Visualizing Mobility Patterns on Networks	
<i>Bram Custers, Wouter Meulemans, Bettina Speckmann, and Kevin Verbeek</i> .....	7:1–7:16
Harmonious Simplification of Isolines	
<i>Arthur van Goethem, Wouter Meulemans, Andreas Reimer, and Bettina Speckmann</i> .....	8:1–8:16
Navigating Your Way! Increasing the Freedom of Choice During Wayfinding	
<i>Bartosz Mazurkiewicz, Markus Kattenbeck, and Ioannis Giannopoulos</i> .....	9:1–9:16
Terrain Prickliness: Theoretical Grounds for High Complexity Viewsheds	
<i>Ankush Acharyya, Ramesh K. Jallu, Maarten Löffler, Gert G.T. Meijer, Maria Saumell, Rodrigo I. Silveira, and Frank Staals</i> .....	10:1–10:16
User Preferences and the Shortest Path	
<i>Isabella Kreller and Bernd Ludwig</i> .....	11:1–11:15

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Versteegen



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Map Matching for Semi-Restricted Trajectories <i>Timon Behr, Thomas C. van Dijk, Axel Forsch, Jan-Henrik Haunert, and Sabine Storandt</i> .....	12:1–12:16
Automated Georeferencing of Antarctic Species <i>Jamie Scott, Kristin Stock, Fraser Morgan, Brandon Whitehead, and David Medyckyj-Scott</i> .....	13:1–13:16

## ■ Preface

This second volume contains the full paper proceedings of the 11th International Conference on Geographic Information Science (GIScience 2021) that was scheduled to be held in Poznań, Poland, 27–30 September 2021, after last year’s conference was postponed due to the widespread outbreak of COVID-19. Given that the pandemic persisted through 2021, the organizing committee decided to hold an online conference instead.

Overall, we received 85 submissions, out of which 29 were full papers and 56 were short papers. While most papers received three reviews, the number of reviews varied between two and four. For the full papers, the review phase was followed by a rebuttal phase in which the authors could react to the reviews and provide clarifications. Next, the reviewers discussed the reviews and rebuttals with a metareviewer, and adjusted their final assessment when appropriate. The metareviewers summarized the reviews and discussion and provided a recommendation to the program chairs. One manuscript was accepted conditionally to undergo another round of editorial checks. In total, we accepted 13 full papers for this second volume. The short papers were in review at the time of compiling this full paper volume.

The accepted papers represent a wide range of topics in GIScience, including work on trajectory and movement analysis, computational geometry, semantics, GeoAI, and agent-based modeling.

The entire GIScience 2021 team would like to express their gratitude to all the authors, reviewers, workshop and tutorial organizers, and anybody else involved in organizing the conference. We are particularly grateful to the emergency reviewers for accepting the increased workload and to the broader GIScience community for providing support, advice, and for their understanding during these difficult times.

The GIScience 2020/21 Organizing Team



## ■ Conference Organization

### **General Chair**

Piotr Jankowski, San Diego State University, USA & Adam Mickiewicz University, Poznań, Poland

### **Program Chairs**

Krzysztof Janowicz, University of California, Santa Barbara, USA  
Judith Versteegen, Wageningen University, The Netherlands

### **Local Organization Chair**

Zbigniew Zwoliński, Adam Mickiewicz University, Poznań, Poland

### **Workshop and Tutorial Chairs**

Grant McKenzie, McGill University, Canada  
Marcin Winowski, Adam Mickiewicz University, Poznań, Poland

### **Sponsorship Chair**

Bernd Resch, University of Salzburg, Austria

### **Publicity Chairs**

Dara Seidl, Colorado Mountain College, USA  
Alfred Stach, Adam Mickiewicz University, Poznań, Poland

### **Arrangements and Logistics**

Joanna Gudowicz, Adam Mickiewicz University, Poznań, Poland  
Robert Kruszyk, Adam Mickiewicz University, Poznań, Poland  
Jarosław Jasiewicz, Adam Mickiewicz University, Poznań, Poland  
Paweł Matulewski, Adam Mickiewicz University, Poznań, Poland  
Małgorzata Mazurek, Adam Mickiewicz University, Poznań, Poland  
Alicja Najwer, Adam Mickiewicz University, Poznań, Poland  
Justyna Weltrowska, Adam Mickiewicz University, Poznań, Poland

### **Metadata Chairs**

Blake Regalia, University of California, Santa Barbara, USA  
Gengchen Mai, University of California, Santa Barbara, USA  
Ling Cai, University of California, Santa Barbara, USA

### **Webmaster**

Jakub Nowosad, Adam Mickiewicz University, Poznań, Poland



## Program Committee

Name	University, Country
Benjamin Adams	University of Canterbury, New Zealand
Ola Ahlqvist	The Ohio State University, USA
Gennady Andrienko	Fraunhofer, Germany
Natalia Andrienko	Fraunhofer, Germany
Clio Andris	Georgia Tech, USA
Andrea Ballatore <sup>m</sup>	Birkbeck, University of London, UK
Kate Beard-Tisdale	University of Maine, USA
Itzhak Benenson	Tel Aviv University, Israel
Michela Bertolotto	University College Dublin, Ireland
Ling Bian	University at Buffalo, USA
Justine Blanford	University of Twente, The Netherlands
Thomas Blaschke	University of Salzburg, Austria
Boyan Brodaric	Geological Survey of Canada, Canada
Chris Brunson	National University of Ireland, Maynooth, Ireland
Ling Cai <sup>e</sup>	University of California, Santa Barbara, USA
Gilberto Camara	INPE, Brazil
Christophe Claramunt	Naval Academy Research Institute, France
Eliseo Clementini	University of L'Aquila, Italy
Tom Cova	University of Utah, USA
Clodoveu Davis	Universidade Federal de Minas Gerais, Brazil
Sytze de Bruin	Wageningen University, The Netherlands
Somayeh Dodge	University of California, Santa Barbara, USA
Suzana Dragicevic	Simon Fraser University, Canada
Matt Duckham	RMIT University, USA
Sara Irina Fabrikant	University of Zurich, Switzerland
Song Gao	University of Wisconsin-Madison, USA
Amy Griffin	RMIT University, Australia
Torsten Hahmann	University of Maine, USA
Jan-Henrik Haunert	Universität Bonn, Germany
Gerard Heuvelink	Wageningen University, The Netherlands
Stephen Hirtle	University of Pittsburgh, USA
Hartwig Hochmair	University of Florida, USA
Yingjie Hu <sup>m</sup>	University at Buffalo, USA
Bernhard Höfle	Heidelberg University, Germany
Marta Jankowska	University of California, San Diego, USA
Bin Jiang	University of Gävle, Sweden
Christopher Jones	Cardiff University, UK
Carsten Keßler	Aalborg University Copenhagen, Denmark
Peter Kiefer	ETH Zurich, Switzerland
Alexander Klippel	Wageningen University, The Netherlands
Jennifer Koch	University of Oklahoma, USA
Christian Kray	University of Münster, Germany
Shawn Laffan	The University of New South Wales, Australia
Tobia Lakes	Humboldt-Universität zu Berlin, Germany

Patrick Laube	Zurich University of Applied Sciences ZHAW, Switzerland
Michael Leitner	Louisiana State University, USA
Wenwen Li	Arizona State University, USA
Arika Ligmann-Zielinska	Michigan State University, USA
Gengchen Mai <sup>e</sup>	University of California, Santa Barbara, USA
Ed Manley <sup>m</sup>	University of Leeds, UK
Bruno Martins <sup>m</sup>	University of Lisbon, Portugal
Grant McKenzie <sup>m</sup>	McGill University, Canada
Mir Abolfazl Mostafavi	Laval University, Canada
Alan Murray	University of California, Santa Barbara, USA
Atsushi Nara <sup>m</sup>	San Diego State University, USA
David O'Sullivan	Victoria University of Wellington, New Zealand
Edzer Pebesma	University of Münster, Germany
Ross Purves	University of Zurich, Switzerland
Martin Raubal	ETH Zurich, Switzerland
Simon Scheider <sup>m</sup>	Utrecht University, The Netherlands
Oliver Schmitz	Utrecht University, The Netherlands
Johannes Scholz	Graz University of Technology, Austria
Johannes Schönig	University of Bremen, Germany
Dara Seidl <sup>m</sup>	Colorado Mountain College, USA
Raja Sengupta	McGill University, Canada
Shih-Lung Shaw	University of Tennessee, USA
Takeshi Shirabe	KTH Royal Institute of Technology, Sweden
Alex Singleton	University of Liverpool, UK
André Skupin	San Diego State University, USA
Seth Spielman	University of Colorado Boulder, USA
Kathleen Stewart	University of Maryland, USA
Martin Swobodzinski	Portland State University, USA
Jean-Claude Thill	University of North Carolina at Charlotte, USA
Sabine Timpf	University of Augsburg, Germany
Martin Tomko	University of Melbourne, Australia
Ming-Hsiang Tsou	San Diego State University, USA
Nico Van de Weghe	Ghent University, Belgium
Marc Van Kreveld	Utrecht University, The Netherlands
Judith Versteegen	Wageningen University, The Netherlands
Monica Wachowicz	RMIT University, Canada
Shaowen Wang	University of Illinois at Urbana-Champaign, USA
Robert Weibel	University of Zurich, Switzerland
John Wilson	University of Southern California, USA
Stephan Winter	University of Melbourne, Australia
Ningchuan Xiao	The Ohio State University, USA
Eunhye Yoo	University at Buffalo, USA
Bailang Yu	East China Normal University, China
Sisi Zlatanova	University of New South Wales, Australia

<sup>e</sup>: (additionally as) emergency reviewer, <sup>m</sup>: (additionally as) meta reviewer



## ■ List of Authors

Ankush Acharyya (10)  
The Czech Academy of Sciences,  
Institute of Computer Science,  
Prague, Czech Republic

Negar Alinaghi  (5)  
Geoinformation, TU Wien, Austria

Timon Behr (12)  
University of Konstanz, Germany

Dominik Bucher  (4)  
Institute of Cartography and Geoinformation,  
ETH Zurich, Switzerland

Bram Custers  (7)  
Eindhoven University of Technology,  
The Netherlands

Anne Driemel  (6)  
Hausdorff Center for Mathematics,  
University of Bonn, Germany

Axel Forsch  (12)  
University of Bonn, Germany

Ioannis Giannopoulos  (5, 9)  
Geoinformation, TU Wien, Austria;  
Institute of Advanced Research in Artificial  
Intelligence (IARAI), Vienna, Austria

Antonia Golab (5)  
Geoinformation, TU Wien, Austria

Carlos Granell  (2)  
Institute of New Imaging Technologies,  
Universitat Jaume I de Castellón, Spain

Jan-Henrik Haunert  (6, 12)  
Institute of Geodesy and Geoinformation,  
University of Bonn, Germany

Herman Haverkort (6)  
Institute of Computer Science,  
University of Bonn, Germany

Barbara Hofer  (2)  
Christian Doppler Laboratory GEOHUM and  
Department of Geoinformatics – Z\_GIS,  
University of Salzburg, Austria

Ye Hong  (4)  
Institute of Cartography and Geoinformation,  
ETH Zurich, Switzerland

Ramesh K. Jallu (10)  
The Czech Academy of Sciences,  
Institute of Computer Science,  
Prague, Czech Republic

Markus Kattenbeck  (5, 9)  
Geoinformation, TU Wien, Austria

Markus Konkol  (2)  
Faculty of Geo-Information Science and Earth  
Observation (ITC), University of Twente,  
Enschede, The Netherlands

Ourania Kounadi  (1)  
Department of Geography and Regional  
Research, University of Vienna, Austria

Isabella Kreller (11)  
University of Regensburg, Germany

Sven Lautenbach  (3)  
Heidelberg Institute for Geoinformation  
Technology (HeiGIT) gGmbH at Heidelberg  
University, Germany

Bernd Ludwig  (11)  
University of Regensburg, Germany

Christina Ludwig  (3)  
GIScience Research Group, Institute of  
Geography, Heidelberg University, Germany

Maarten Löffler (10)  
Department of Information and Computing  
Sciences, Utrecht University, The Netherlands

Henry Martin  (4)  
Institute of Cartography and Geoinformation,  
ETH Zurich, Switzerland;  
Institute of Advanced Research in Artificial  
Intelligence (IARAI), Austria

Bartosz Mazurkiewicz  (9)  
Geoinformation, TU Wien, Austria

David Medyckyj-Scott  (13)  
Manaaki Whenua Landcare Research,  
Auckland, New Zealand

Gert G.T. Meijer (10)  
Academy of ICT and Creative Technologies,  
NHL Stenden University of Applied Sciences,  
The Netherlands

Wouter Meulemans  (7, 8)  
Eindhoven University of Technology,  
The Netherlands

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.  
Editors: Krzysztof Janowicz and Judith A. Verstegen



Leibniz International Proceedings in Informatics  
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Fraser Morgan  (13)  
Manaaki Whenua Landcare Research,  
Auckland, New Zealand
- Daniel Nüst  (2)  
Institute for Geoinformatics,  
University of Münster, Germany
- Frank O. Ostermann  (2)  
Faculty of Geo-Information Science and Earth  
Observation (ITC), University of Twente,  
Enschede, The Netherlands
- Fiona Polzin (1)  
ITC-Faculty of Geoinformation and Earth  
Observation, University of Twente, Enschede,  
The Netherlands
- Martin Raubal  (4)  
Institute of Cartography and Geoinformation,  
ETH Zurich, Switzerland
- Andreas Reimer (8)  
Eindhoven University of Technology,  
The Netherlands
- Peter Rottmann (6)  
Institute of Geodesy and Geoinformation,  
University of Bonn, Germany
- Heiko Röglin (6)  
Institute of Computer Science,  
University of Bonn, Germany
- Maria Saumell (10)  
The Czech Academy of Sciences, Institute of  
Computer Science, Prague, Czech Republic;  
Department of Theoretical Computer Science,  
Faculty of Information Technology, Czech  
Technical University in Prague, Czech Republic
- Eva-Marie Schömann (3)  
GIScience Research Group, Institute of  
Geography, Heidelberg University, Germany
- Jamie Scott  (13)  
Massey University, Auckland, New Zealand
- Rodrigo I. Silveira (10)  
Department of Mathematics,  
Universitat Politècnica de Catalunya,  
Barcelona, Spain
- Bettina Speckmann  (7, 8)  
Eindhoven University of Technology,  
The Netherlands
- Frank Staals (10)  
Department of Information and Computing  
Sciences, Utrecht University, The Netherlands
- Kristin Stock  (13)  
Massey Geoinformatics Collaboratory,  
Massey University, Auckland, New Zealand
- Sabine Storandt (12)  
University of Konstanz, Germany
- Thomas C. van Dijk  (12)  
University of Bochum, Germany
- Arthur van Goethem (8)  
Eindhoven University of Technology,  
The Netherlands
- Kevin Verbeek  (7)  
Eindhoven University of Technology,  
The Netherlands
- Brandon Whitehead  (13)  
Manaaki Whenua Landcare Research,  
Auckland, New Zealand
- Yanan Xin  (4)  
Institute of Cartography and Geoinformation,  
ETH Zurich, Switzerland
- Alexander Zipf  (3)  
GIScience Research Group, Institute of  
Geography, Heidelberg University, Germany;  
HeiGIT gGmbH at Heidelberg University,  
Germany

# Adaptive Voronoi Masking: A Method to Protect Confidential Discrete Spatial Data

Fiona Polzin 

ITC-Faculty of Geoinformation and Earth Observation,  
University of Twente, Enschede, The Netherlands

Ourania Kounadi<sup>1</sup>   

Department of Geography and Regional Research, University of Vienna, Austria

---

## Abstract

Geomasks assure the protection of individuals in a discrete spatial point data set by aggregating, transferring or altering original points. This study develops an alternative approach, referred to as Adaptive Voronoi Masking (AVM), which is based on the concepts of Adaptive Aerial Elimination (AAE) and Voronoi Masking (VM). It considers the underlying population density by establishing areas of K-anonymity in which Voronoi polygons are created. Contrary to other geomasks, AVM considers the underlying topography and displaces data points to street intersections thus decreasing the risk of false-identification since residences are not endowed with a data point.

The geomasking effects of AVM are examined by various spatial analytical results and are compared with the outputs of AAE, VM, and Donut Masking (DM). VM attains the best efficiency for the mean centres whereas DM does for the median centres. Regarding the Nearest Neighbour Hierarchical Cluster Analysis and Ripley's K-function, DM demonstrates the strongest performance since its cluster ellipsoids and clustering distance are the most similar to those of the original data. The extend of the original data is preserved the most by VM, while AVM retains the topology of the point pattern. Overall, AVM was ranked as 2<sup>nd</sup> in terms of *data utility* (i) and also outperforms all methods regarding the *risk of false re-identification* (ii) because no data point is moved to a residence. Furthermore, AVM maintains the *Spatial K-anonymity* (iii) which is also done by AAE and partly by DM. Based on the performance combination of these factors, AVM is an advantageous technique to mask geodata.

**2012 ACM Subject Classification** Security and privacy → Privacy protections; Security and privacy → Data anonymization and sanitization; Information systems → Geographic information systems; Mathematics of computing → Exploratory data analysis

**Keywords and phrases** Geoprivacy, location privacy, geomasking, Adaptive Voronoi Masking, Voronoi Masking, Adaptive Aerial Elimination, Donut Geomasking, ESDA

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.1

**Supplementary Material** The Geoprivacy Github repository contains the scripts to run AVM (Also AAE) as well as one of the six area data sets used in this study:

*Software:* <https://github.com/okounadi/Geoprivacy>

## 1 Introduction

### 1.1 Background

The advances of GIS and the interest in spatial analysis have led to an increase of thematic maps in research and online platforms visualizing point data. However, several studies in health geography, reproductive and sexual health did not anonymize or aggregate data; instead, the original data were used [4, 14, 17]. Publishing an individual's location either in

---

<sup>1</sup> Corresponding author



paper or digital form - knowingly or unknowingly - increases the risk of re-identification by and violates individual privacy. How simple the re-identification of individuals is, was already demonstrated by Brownstein et al. [4]: By applying the reverse-identification method, 7% of the spatially coded addresses were accurately identified while all 550 of the plotted address points were disclosed within 14 m of the right address. Furthermore, Kounadi and Leitner [17] exposed that within an eight-year duration, almost 70,000 home addresses had been disclosed in academic research.

The consequences of disclosure are vast; an individual being identified as an HIV-patient - correctly or wrongly - can affect him or her by discrimination or social stigmatization [27]. Identifications may cause harassment [9], unwanted advertisement or humiliation [26, 25]. Kounadi and Leitner [19] criticize that general rules on privacy do not include details of the spatial re-identification risk notwithstanding the fact that relevant research and reports on geodata exist [12, 20]. Consequently, confidential spatial data sets do not only have to be preserved but also need to comply with present-day restrictions and regulations on the right to privacy [19]. However, Ajayakumar et al. [1] criticized that geomasks are still unavailable for many institutions due to the lack of expertise in geospatial proficiency although the awareness of the power of mapping has grown particularly in health organizations and clinics which have become spatially literate lately. The authors stress that geomasks need to become more of a real-world requirement.

## 1.2 Problem statement

Some geomasks displace the points a specific distance aside from its original location (e.g., *local random rotation* by Leitner and Curtis [22] and *Voronoi masking* (VM) by Seidl et al. [29]), while others aggregate points (e.g., *spatial and point aggregation* by Armstrong et al. [3]). Other geomasks consider the underlying population density adapting the displacement error such as the *Donut Geomasking* (DM) [15] and the *Adaptive Areal Elimination* (AAE) [19]. By considering the population density, the “masker” is able to determine a level of *K-anonymity* in which each record (i.e. person) within a masked data set cannot be identified from at least K-1 records [24]. Regarding geodata, K-anonymity assures that every location such as household, address or an individual’s location cannot be differentiated from minimum K-1 locations. This means, that *spatial K-anonymity* (SKA) describes the probability of identifying a location that can be linked to an individual by reverse geocoding. This is needed to evaluate the degree of privacy and when measuring the degree of displacement.

A possible solution to prevent re-engineering of original locations could be points’ aggregation. However, when doing so, the ability to distinguish spatial relations or clusters and deriving persuasive information is decreased [30, 3, 21]. Obviously, the data becomes less useful for research purposes [30, 13]. Contrary to aggregation, geomasks that modify the locations are preferred for analytical purposes. Nevertheless, the transferred points can be moved to a position which has real observations [23] or where they cannot exist [7], resulting in *false identification* [29]. False identification represents the incorrect linking of a household or person to a data point. Contrary to that, correct identification is the correct linkage of a household or person to a data point [29]. The consequences of identification can result in negative effects impeding an individual’s social prominence [13]. Besides, it can unintentionally involve individuals, who were not part of the research [7]. Such limitations influence both the disclosure risk and a successful investigation of spatial patterns.

Generally, there is neither a recommended nor approved geomask technique [30, 13] and each method has disadvantages and advantages. Zandbergen [30] suggests counterbalancing data utility and confidentiality protection. Also, not a lot of geomasks consider the underlying

topography except for the *Street aggregation at intersection* or at *midpoint* [22] or the *Location Swapping* method [31]. Yet these that do consider the underlying topography do not offer a predefined level of SKA. It is evident that existing techniques must be improved to overcome such shortcomings and also become widely accessible.

### 1.3 Study scope and design

Our alternative approach, referred to as Adaptive Voronoi Masking (AVM), is based on the concepts of AAE and VM. AVM shall protect the individual's privacy based on SKA while also decreasing the false re-identification risk. We evaluate known geomasks, namely the VM, the AAE, and the DM, and compare them with the proposed AVM in terms of three key aspects: a) SKA, b) false re-identification, and c) data utility.

In the next section (Methodology) we explain the two geomasks that AVM is based on (VM and AAE) and then describe the functionality of AVM. Next, we present the exploratory spatial data analysis (ESDA) methods that are used to compare and evaluate the original data points with the outcome of the geomasks (i.e. masked data points). Last, we introduce the study area, the software, and data used. In section 3 (Results), we report the ESDA results and finally discuss and conclude our findings in section 4 (Conclusion). Apart from the AVM, DM, and AAE, we also evaluate the DM geomask. DM was chosen as a comparative geomask since it is a popular technique and it has a small effect on the geographical characteristics of the original point pattern as highlighted in academic literature [15, 30, 2]. The algorithm for this method was retrieved online<sup>2</sup>.

## 2 Methodology

### 2.1 Adaptive Areal Elimination (AAE)

AAE assures privacy by moving the original locations within uncertainty areas. The so-called uncertainty areas describe an area, where the masked points are displaced in, e.g. torus or circle [19]. For instance, DM moves the original data within an uncertainty area selected from a uniform distribution [15] while the population-density-based Gaussian spatial blurring dislocates points within a circle based on a normal distribution [5]. However, these geomasks assume that population is homogeneously distributed - which is not the case in most instances. This assumption can result in masked data points with a lower actual K-anonymity than the estimated K-anonymity [2]. Hence, AAE is aiming to ensure K-anonymity even when the geomasking method and its parameters are known. K-anonymity can be measured precisely when uncertainty areas do not overlap and when it is applied at a lower or equal level of the available resolution [19].

To execute the AAE algorithm, two data sets are needed: a) a point file and b) a spatial data set that either includes an attribute with discrete information (e.g., as administrative units containing an attribute field with the total households in each unit) or represents discrete information (e.g., point data representing households). This attribute is called RoRi (risk of re-identification). Generally, risk of re-identification can contain information such as addresses, households, or population. A disclosure value for this field predefined to describe the minimum K-anonymity which is used to obscure confidential information. In the next step, the process of merging polygons starts: depending on the disclosure value,

---

<sup>2</sup> [https://mserre.sph.unc.edu/BMElab\\_web/donutGeomask/donutGeomask.htm](https://mserre.sph.unc.edu/BMElab_web/donutGeomask/donutGeomask.htm) (Last accessed on January 22<sup>th</sup>, 2021)

## 1:4 Adaptive Voronoi Masking (AVM)

every polygon containing a lower risk of re-identification value than the disclosure value, is merged with its neighbouring polygon or polygons until each polygon has values that are either greater than or equivalent with the disclosure value to create the K-anonymized areas. Next, original data are aggregated to the centroids of the merged polygons or randomly displaced within the merged polygons. Random displacement can be performed by a random perturbation to the coordinates of each data point by a random distance and at a random direction (equation 1).

$$\begin{aligned} X_m &= X_o + D * \cos(\Theta) \\ Y_m &= Y_o + D * \sin(\Theta) \end{aligned} \tag{1}$$

where,  $X_o, Y_o$  are the original coordinates of a point,  $X_m, Y_m$  are the resulting masked coordinates,  $D$  is a random value within a predefined range, and  $\Theta$  is a random angle.

In the AAE each masked point shall lie within its k-anonymized polygon. Thus, the displaced masked point/s has to be conditioned on the boundaries of each polygon. In this study, we implement the random displacement that yielded better performance results in the study by Kounadi and Leitner (2016). When studying the outputs of AAE more closely, some masked data are moved further distances than necessary. This can be explained by the process of merging polygons that selects the neighbour with the longest boundary, which may result in K-anonymized areas that are larger than needed to ensure SKA.

## 2.2 Voronoi Masking (VM)

VM creates Voronoi polygons around the original data points are displaced to the closest segment (edge) of its corresponding polygon [29, 13]. The theoretical basis of creating Voronoi polygons starts with the triangulation of the original points into an irregular network that meets the *Delaunay criterion* (i.e. no point is inside the circumcircle of any triangle). Then, the perpendicular bisectors for each triangle edge are generated. These are the edges of the Voronoi polygons while the locations of the bisector's intersections determine the vertices. Every point within each polygon is closer to the original point of its creation than to other original points.

Advantages of VM is that points in neighbouring polygons are displaced to the same position, enhancing their K-anonymity and that a higher point density results in smaller distances between the original data and masked data thus giving a pattern that is similar to the original one [29, 13]. For a small scale area or an area with a minimum of two households, VM dislocates the original data a lesser distance than compared to other geomasks that do not consider the underlying settlement patterns. VM is an efficient approach regarding the preservation of the spatial point pattern as it has been proved by Seidl et al. [29] who implemented various methods to evaluate its performance. Finally, Seidl et al. [29] praise that in case of applying a data set that is including all residences within the area of interest, no displaced point will be located on an actual residence and thus false identification of residences is not possible. Points are typically located in the centre of a parcel or at the street segment. VM will definitely move points away from these locations. However, this process does not guarantee that segments of Voronoi polygons will not cross residential parcels and therefore VM cannot decrease the risk of false identification in this regard.

When applying VM in areas with scattered residences some data points will be dislocated at large distances, which affect spatial patterns. Also, a smaller amount of masked data will be depicted on the map than the original data due to the overlapping of points at the

displaced locations. Although this assures a higher K-anonymity, the map viewer may not be aware that some points represent at least two addresses, increasing the risk of spatially analysing or perceiving the output differently. Last, although K-anonymity is increased, compared to other geomasks, a predefined level of SKA cannot be guaranteed.

### 2.3 Adaptive Voronoi Masking (AVM)

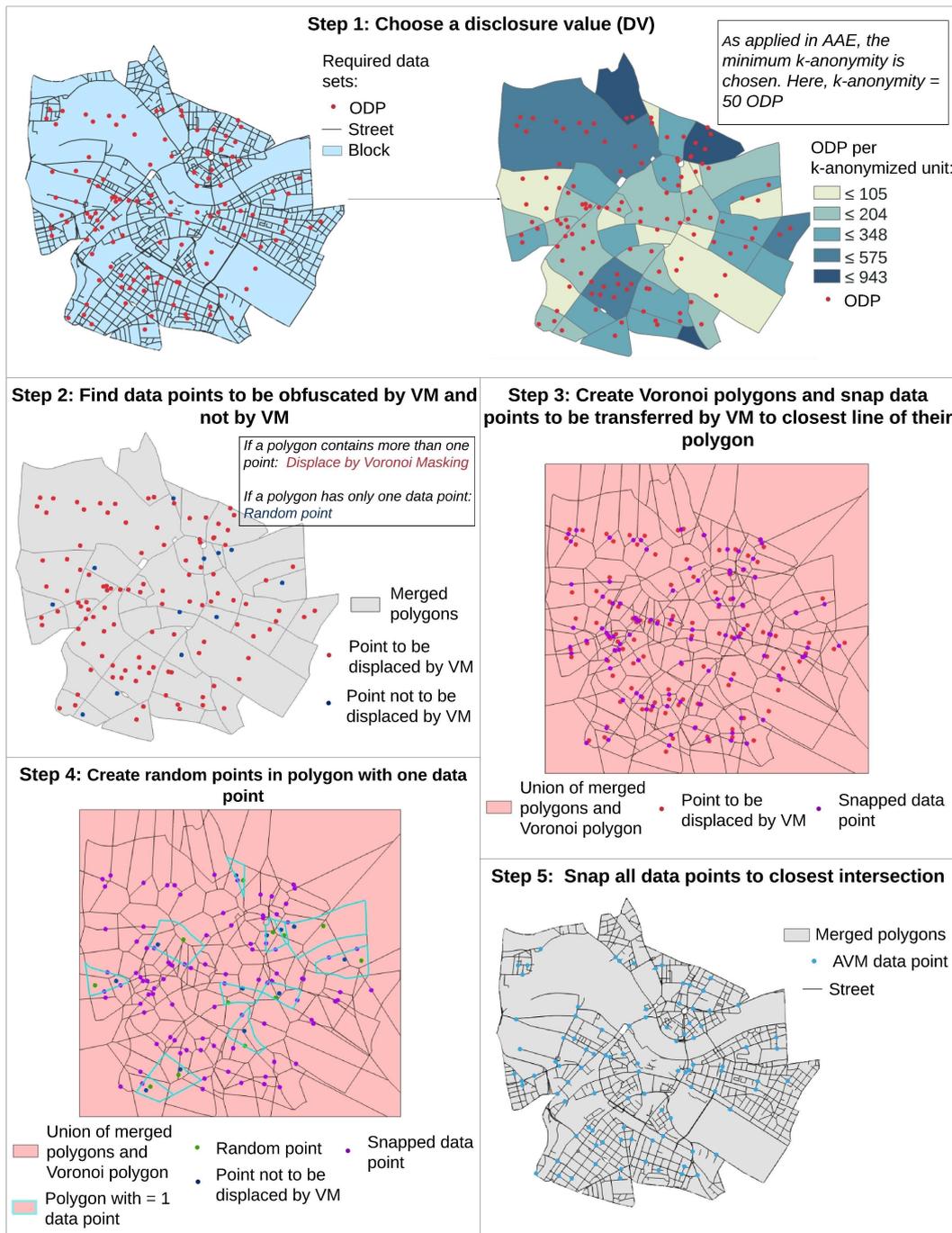
AVM extracts the asset of considering the underlying population density by joining polygons as AAE does and displaces the original data based on the concept of VM. In respect thereof, the original data are moved to the closest segment of their corresponding Voronoi polygon which lies within their merged AAE-polygon. In case a Voronoi segment lies outside its dissolved polygon, the point is transferred to the boundary of the merged polygon and not to the edge of the Voronoi polygon. Through that, AVM intends to circumvent the predicament of moving points to a polygon containing a different population threshold thus preserving the *predefined SKA*. Further, the underlying topography is considered by moving points to the closest street intersection that has a higher amount of surrounding buildings than if moved to the nearest segment. Through that, AVM avoids shifting the points directly to another residence causing *false re-identification* but it also prevents the displacement to *invalid locations* such as water bodies or forests.

To execute AVM, the following data sets are required: a) a point file (as needed in VM and AAE), b) a polygon file including risk of re-identification information (as required in AAE), and c) a line file depicting the street network. Firstly, the data is pre-processed as done for AAE. Subsequently, a disclosure threshold for the risk of re-identification field is selected and polygons with a smaller value than the chosen disclosure value are merged with its adjacent polygon until all polygons receive a value that is greater or equal to the set disclosure value. Here, the general spatial rule is applied defining that every polygon is combined with the bordering polygon that has the longest shared border [19].

Secondly, every data point that is lying within a polygon with at least two data points is transferred by the concept of the VM technique. It is guaranteed that the data points are replaced to the closest segment of their corresponding Voronoi polygon within their dissolved polygon. Thirdly, the polygons containing only one data point randomly transfer the data point within their merged polygon. Afterwards, all newly displaced points are shifted to the closest street intersection inside their K-anonymized polygon. Figure 1 shows the outputs of the steps using as an example the city centre of Dresden, Germany.

### 2.4 ESDA for evaluating geomasking performance

ESDA identifies and characterizes locations, shapes, and magnitudes of statistically substantial patterns within an area of interest [10]. Studies on geoprivacy implemented ESDA methods on original data and masked data to investigate and compare the performance of geomasks. Armstrong et al. [3] scrutinized the effect of geomasks by exploring pair-wise relations, event-geography relations, anisotropies, and trends. Seidl et al. [29] applied the kernel density estimation, global Moran's I, distance to K-nearest neighbour, the cross K-function - also known as Ripley's K-function, and the nearest neighbour hierarchical cluster analysis. Kwan et al. [21] also applied the Ripley's K-function, the kernel density estimation, and examined the visualisation of the point pattern. Leitner and Curtis [22] analysed the visualisation of the point pattern as well. Several approaches exist to analyse the efficiency of geomasks. Here, we use four methods that were already used in previous studies and are described in the next subsections.



■ **Figure 1** A visualisation of the AVM outputs in the city centre of Dresden.

### 2.4.1 Visualisation of point pattern

This technique is used to a) scrutinize the extent of the original data and compare it with that of the masked data and b) to investigate whether the masked data are displaced on other residences increasing the risk of false re-identification or are transferred to void locations such as forests or lakes.

### 2.4.2 Central tendency

The mean and median centres of the original data and the masked data are compared through their distance's divergence. This has been applied by Seidl et al. [29] and Gupta and Rao [13].

### 2.4.3 Ripley's K function

Ripley's K-function identifies whether the masked points are clustered, dispersed, or randomly distributed and whether the point distribution between original data and masked data remains linked or not. In the case of linked point distribution, the geomasks perform spatially dependent on the original data. Ripley's K-function conflates spatial dependence regarding point feature scattering or aggregation over a variety of distances [8], which returns a more detailed output than other ESDA pattern detection techniques. By analysing the spatial patterns over several lengths as well as spatial scales, the point patterns alter. Thus, it can reflect how the scattering or aggregating of points centroids shifts when the size of the neighbourhood varies.

### 2.4.4 Nearest neighbour hierarchical cluster analysis

In most previous studies, the impact of geomasks on original hot spots has been probed. This is important since clustering detection plays a vital role in spatial analysis. For instance, by detecting hot spots, high concentrations of crime incidents can be explored and predicted for future scenarios [6]. Nearest neighbour hierarchical cluster analysis allows examining and comparing the clustering pattern of the original data with the pattern of the masked data regarding amount of clusters, size, orientation, density.

## 3 Experiments' settings

### 3.1 Study area

The choice of the study area is based on the availability of processed and free data. Moreover, area data sets must allow different levels of spatial granularity and population density. The chosen area is the Free State of Saxony in Eastern Germany. Saxony has 13 districts containing more than 4 million inhabitants<sup>3</sup>, of which more than 563,000 were registered in the state capital Dresden. Yet, the highest population and population density are found in the city of Leipzig with a total of 587,857 people and 1,974 inhabitants per km<sup>2</sup>. Contrary to that, the district Nordsachsen has only 97 inhabitants per km<sup>2</sup> - the lowest in Saxony. Hence, the State of Saxony is an explicit choice to investigate the performance of geomasks because it has highly populated as well as rural areas. The geomasks are applied on the State of Saxony, the city of Leipzig because it has the most inhabitants and the highest population density, and the district of Zwickau. Zwickau was chosen because when calculating the average inhabitants (ca. 313,685) and population density (ca. 493/km<sup>2</sup>) per district in Saxony, Zwickau has the closest values (inhabitants: 317,531; population density: 334/km<sup>2</sup>).

---

<sup>3</sup> <https://www.statistik.sachsen.de/> (Last accessed on January 22<sup>th</sup>, 2021)

### 3.2 Data

For the polygon file, a line shapefile representing the street network in Saxony was derived online<sup>4</sup>, and used to create streets blocks. It was aimed to develop blocks which are not too coarse but also not too small. Since the original road network file also included several street classes such as “footway”, “path”, or “cycleway”, all street duplicates, as well as all street classes except for “primary”, “secondary”, and “tertiary”, were deleted off the shapefile to generalize the road network. Thus, an enormous amount of small blocks is avoided resulting in shorter processing times of the geomasks. Also, only single-line road features in place of matched pairs of divided road lanes were maintained. Small and open configurations of roads were removed. Next, polygon features were created from the remaining polylines. Any polygons outside the study area were removed.

The same street network dataset was also used for intersection displacement step of AVM. However, in this case, it was more detailed containing also the “residential class”. Thus, a smaller but yet meaningful displacement of the data points can be achieved. More street classes such as “footway” or “cycleway” were not included to prevent false re-identification.

Also, additional attributes were included. First, risk of re-identification information needs to be added within the polygons. Hence, a point data set with addresses in Saxony from 2018 was chosen and can be downloaded directly from ESRI<sup>5</sup>. Originally, the point data set consists of 947,164 data points. The points were counted per polygon as the risk of re-identification information. Second, two “sensitive” data sets were created as random subsets from the addresses in Saxony consisting of 2,000, and 200 points for each and thus mimicking different population densities to examine the performance of AVM at different situations. These data sets simulate potential confidential or private discrete data. Third, attributes such as “id” (unique identifier) and “area” (size of a polygon) were added as they are necessary for the algorithms. Finally, the polygons were clipped based on the boundaries of the three study areas. The boundaries were obtained from the *Federal Agency for Cartography and Geodesy*<sup>6</sup>. The clipped study areas do not completely correspond in size with the original district of Zwickau, City of Leipzig, or the Free State of Saxony. This is due to the removal of smaller streets creating somehow different sizes and shapes of the study areas. In the case of referring to a specific study area with a certain number of points, the data sample is named *study area + number of points* (i.e., Saxony 200). Figure 2 shows the resulting area data sets.

### 3.3 Software

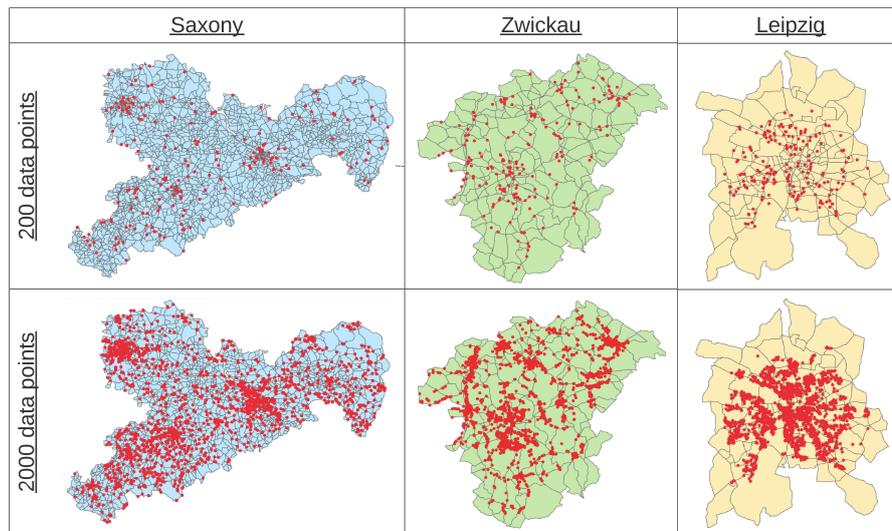
We used ArcGIS Pro 2.5 by the international GIS-software developer ESRI and CrimeStat 3.3. by Levine & Associates (2020). ArcGIS Pro is used for data exploration, visualisation, for running the AAE and DM algorithms, and for the creation of the AVM and VM algorithms. Hereby, the embedded ArcPy Python package was used. The ESDA evaluation methods, with the exception of nearest neighbour hierarchical cluster analysis, were operated in ArcGIS Pro. Nearest neighbour hierarchical cluster analysis was performed in CrimeStat 3.3 (i.e. a program of spatial statistics for exploring locations of crime incidents). CrimeStat can be downloaded for free online<sup>7</sup>.

<sup>4</sup> <https://download.geofabrik.de/europe/germany/sachsen.html> (Last accessed on January 22<sup>th</sup>, 2021)

<sup>5</sup> <https://opendata-esri-de.opendata.arcgis.com/datasets/esri-de-content::adressen-sachsen> (Last accessed on January 22<sup>th</sup>, 2021)

<sup>6</sup> <https://www.bkg.bund.de/DE/Home/home.html> (Last accessed on January 22<sup>th</sup>, 2021)

<sup>7</sup> <https://www.icpsr.umich.edu/CrimeStat/> (Last accessed on January 22<sup>th</sup>, 2021)



■ **Figure 2** The six area data sets that are used in the study.

## 4 Results

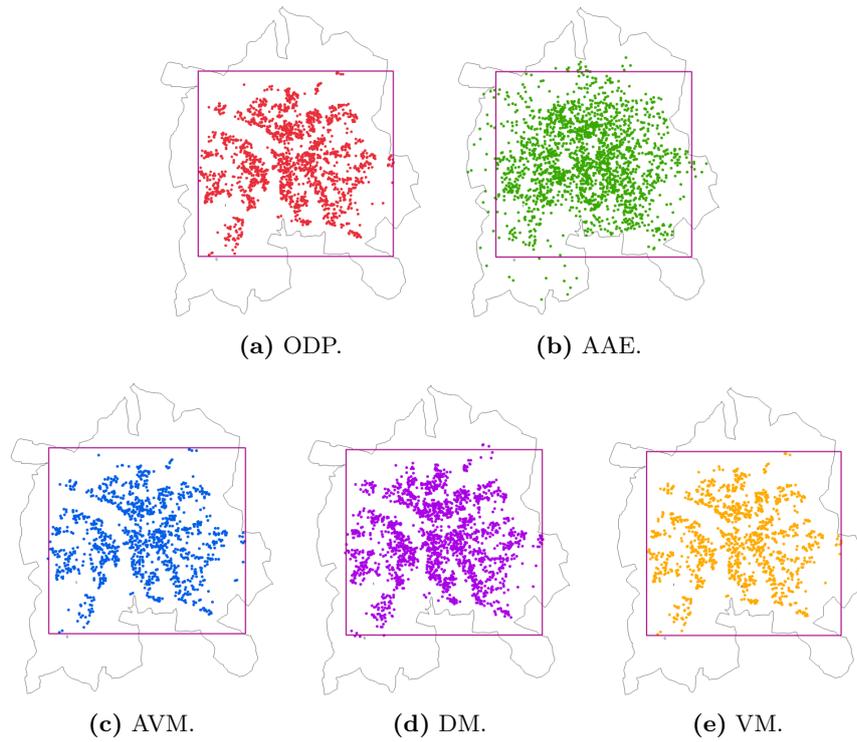
AVM, AAE, and DM were applied with a SKA level of 50 addresses. VM is not an adaptive geomask and thus a SKA level cannot be predefined and guaranteed. The four ESDA methods are applied to the original data as well as on the masked data to examine the effects of the geomasks on the original data. [18] and detect dissimilarities of spatial information loss and the preservation of original data granularity [28]. In the ideal case, the spatial analysis of the AVM masked data will be equal to that of the original data.

### 4.1 Visualisation

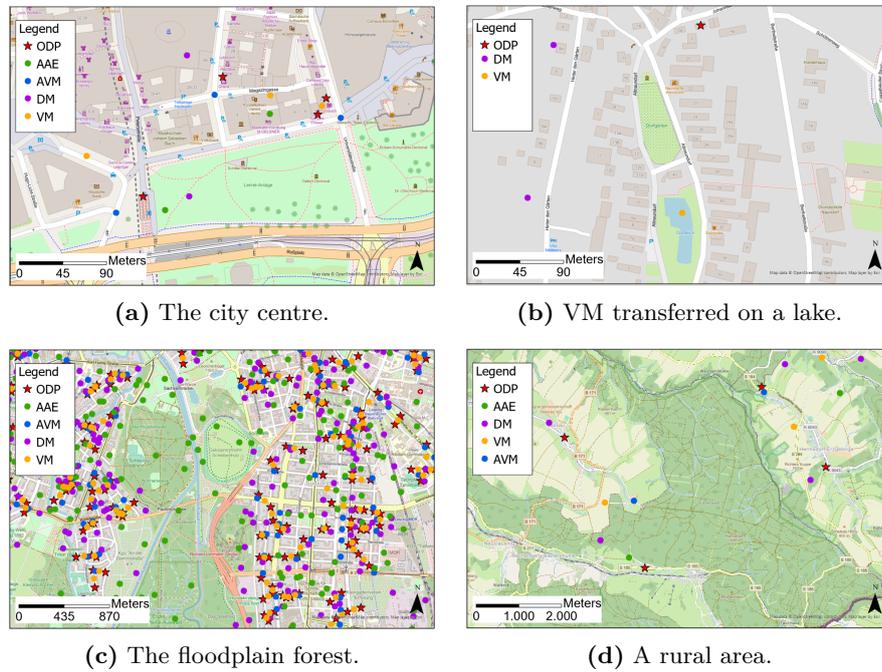
Figure 3 shows the extent of the original data for Leipzig 2000. AAE preserves the spatial extent of the original data the least. For Leipzig 2000, 53 points were dislocated outside of the original extent. DM performs more successfully than AAE with only five points not being located within the extent of the original data. The new technique AVM retains the spatial extent more effectively than AAE and DM. For Leipzig 2000 all points were within the extent while in the other area data sets only one to three points were located outside the extent. VM is outperforming the other geomasks regarding the preservation of the original extent. Only one data set (Leipzig 200) has one data point outside the original extent. We created the same maps for each area data set but since the observations are similar other maps are not presented. By ranking the performance of the geomasks, we can see that VM preserves the extend of original data the most (1<sup>st</sup>), followed by AVM (2<sup>nd</sup>), then DM (3<sup>rd</sup>), and last is AAE (4<sup>th</sup>).

Regarding the preservation of the point pattern, it is perceivable that AAE seems to abandon the pattern the most: Particularly Figure 4b shows that the strongly visible floodplain forest, which intersects the city of Leipzig from northwest to southwest, is not kept by AAE. Unlike that, AVM, DM, and VM maintain this meandering space in most regard while AAE blurs this region completely. Figure 4a is illustrating a park in the city centre of Leipzig. All original data originate on buildings. Every geomask except for AVM displaced data points to either buildings or an uninhabited area as here, the park. Only one

1:10 Adaptive Voronoi Masking (AVM)



■ **Figure 3** The bounding box shows the extent of the original data in Leipzig, while inner points are the masked data.



■ **Figure 4** The results of the geomasks (masked data) compared to the original data in various locations.

VM point had been transferred to a street. Contrary to that, AVM moved all data points to a street intersection. Figure 4b is depicting an area with family homes, a central park, and a lake with a displaced VM data point. Another DM data point had been moved on a house. Figure 4c portrays the floodplain forest in Leipzig. It is noticeable that many points transferred by AAE are laying within this natural habitat which is invalid for a displacement location. Finally, the last Figure 4d show rural areas within Saxony with a low point density. As in the previous examples, AVM points remain on street intersections decreasing the risk of false identification. Again, AAE, DM, and VM moved data points to uninhabited areas, i.e. a forest or grassland. In this analysis, AVM is ranked as 1<sup>st</sup> in not translating points to illogical locations or false residencies, while all other geomasks follow.

## 4.2 Central tendency

The displacement distance between the masked and original mean and median centres are shown in table 1. Regarding the mean centres, VM and DM outperform AAE and AVM in most obfuscation settings. DM has the smallest displacement distances for Leipzig 200 (12.71m), Zwickau 200 (4.53m), Zwickau 2000 (3.97m), and Saxony 200 (98.91m). VM has the smallest displacement distances for Leipzig 2000 (1.18m) and Saxony 2000 (9.61 m). The AAE encompasses the furthest displacement distances for all data sets, while AVM performs better yet worse than DM and VM.

Regarding the median centres, VM surpasses the other geomasks. It has a displacement distance of 1.23m to the original median centre of Leipzig 2000, 42.33m to the one of Zwickau 200, and 13.93m to the one of Saxony 2000. DM presents the lowest displacement distances for Leipzig 200 (1.63m) and Saxony 200 (90.82m), while AVM has the smallest displacement for Zwickau 2000 (1.61m). Again, AAE has much greater displacement distances than the other three geomasks (except for Saxony 200).

Finally, it can be seen that the lower the point density, the stronger the variation of the geomasks' results. For instance, the 200 data points in Saxony represent the strongest variations and the highest displacement distances between the masked and original mean and median centres. Contrary to that, Leipzig 2000 has the highest population density and demonstrates the smallest displacement distances and lowest variations between the mean and median centres of the masked data and original data. The geomasks' performance was ranked for each area data set and then the mode of the rank was derived. For the mean displaced distance, DM yields the closest value to the original data mean (1<sup>st</sup>), followed by VM (2<sup>nd</sup>), then AVM (3<sup>rd</sup>), and last is AAE (4<sup>th</sup>). For the median displaced distance VM yields the closest value to the original data median (1<sup>st</sup>), followed by DM (2<sup>nd</sup>), then AVM (3<sup>rd</sup>), and last is AAE (4<sup>th</sup>).

## 4.3 Ripley's K-function

We calculated the expK-values and obsK-values of the original data and masked data of every five bands. The dissimilarity here is calculated as the clustering distance divergence of the masked data from the original data and it is shown in Table 2. For all area data sets and bands, the original data and masked data attain higher obsK-values than the expK-value indicating a strong clustering pattern. A second observation is that dissimilarities are more distinct in smaller bands than in larger bands. The third observation is that VM and AVM create a more clustered pattern than the original (shown by mostly negative divergence values), while AAE and DM tend to create a less clustered pattern than the original (indicated by mostly positive values).

## 1:12 Adaptive Voronoi Masking (AVM)

■ **Table 1** Displacement distances (in meters) from the mean/ median centre of the original data to the mean/ median centres of the masked data. The lowest displacement distance per data set is depicted in bold. The method that scores most times the best is DM, followed by VM.

Area data set	AVM		VM		AAE		DM	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Leipzig 200	28,86	28,11	12,72	40,3	129,72	158,17	<b>12,71</b>	<b>1,63</b>
Leipzig 2000	3,57	5,51	<b>1,18</b>	<b>1,23</b>	44,24	31,58	3,02	3,12
Zwickau 200	85,71	55,22	23,48	<b>42,33</b>	177,18	562,56	<b>4,53</b>	100,68
Zwickau 2000	10,11	<b>1,61</b>	6,14	3,39	39,46	241,07	<b>3,97</b>	8,27
Saxony 200	616,41	1528,4	182,45	231,68	766,97	1357,64	<b>98,91</b>	<b>90,82</b>
Saxony 2000	27,46	65,52	<b>9,61</b>	<b>13,93</b>	39,83	66,85	14,39	29,08

■ **Table 2** Ripley's K results on the divergence of the clustering distance (in meters) of the masked data from the original data, from 99 simulations for each area data set and in three bands. The smallest divergence is marked in bold. DM yielded the smallest values most times.

Area data-set Geomask /bands	Leipzig 200			Leipzig 2000			Zwickau 200		
	1-5	6-10	11-15	1-5	6-10	11-15	1-5	6-10	11-15
AVM	-847	-230	64	-361	<b>-61</b>	-47	-1650	-750	-342
VM	-1857	-1519	-680	-351	451	<b>-39</b>	-3279	-2296	-1775
AAE	1113	664	695	543	858	977	1271	2226	2593
DM	<b>525</b>	<b>77</b>	<b>32</b>	<b>183</b>	132	116	<b>371</b>	<b>443</b>	<b>272</b>
Area data-set Geomask /bands	Zwickau 2000			Saxony 200			Saxony 2000		
	1-5	6-10	11-15	1-5	6-10	11-15	1-5	6-10	11-15
AVM	-466	-198	-134	-454	642	1529	<b>-197</b>	127	136
VM	<b>9</b>	<b>22</b>	<b>-43</b>	-13443	-10122	-6429	-2138	-1091	-538
AAE	1413	1495	1854	2722	-276	<b>301</b>	1835	1383	1090
DM	753	410	379	<b>-251</b>	<b>202</b>	399	375	<b>53</b>	<b>34</b>

DM has the most similar obsK-values for Leipzig 200 (all bands), Saxony 200 (bands one to ten), Saxony 2000 (bands six to fifteen), and Zwickau 200 (all bands). AVM scores the most alike obsK-values for Leipzig 2000 (bands six to ten) and Saxony 2000 (bands one to five). VM displays a stronger point accumulation than the original data for data sets with a low point density (Leipzig 200, Zwickau 200, and Saxony 200) and demonstrates the most dissimilar obsK-values than the original data which can be elucidated by the technique's character to alter the point pattern in scattered areas due to greater displacement distances of the established Voronoi polygon. However, VM reaches very similar obsK-values to those of the original data for Leipzig 2000 (bands eleven to fifteen), and Zwickau 2000 (all bands). Again, AAE fares the weakest results.

We derived the geomasks' ranks for each area data set/ band and then the mode rank. DM is ranked 1<sup>st</sup>, followed by AVM (2<sup>nd</sup>), then VM (3<sup>rd</sup>), and last is again AAE (4<sup>th</sup>).

### 4.4 Nearest neighbour hierarchical cluster analysis

Table 3 shows the number of clusters, the mean points per cluster, and mean cluster density per m<sup>2</sup> for each original data and masked data and in each area dataset. For Leipzig 200, the original data produced six clusters, with a mean of 7.33 points per cluster. Regarding

the number of clusters, AAE, AVM, and DM are the nearest to the original data value with seven clusters. AVM has the closest mean points per cluster at 7.50 and the most similar cluster density. For Leipzig 2000, VM has the closest number of clusters as the original data (VM: 139; original data: 136). Regarding the mean points, original data contains 7.73 points and DM reaches the nearest measure at 7.83. Also, DM has the most alike density at  $0.0000979 \text{ m}^2$  (original data:  $0.0001144 \text{ m}^2$ ).

About Zwickau 200, the original data yielded eight clusters, mean points at 7.75, and a density of  $0.0000055 \text{ per m}^2$ . Regarding the first metric, DM has the same value. Regarding mean points, VM outperforms the other geomasks at 7.82, whilst DM shows the nearest value for the cluster density at  $0.0000060 \text{ per m}^2$ . For Zwickau 2000, the original data yielded 110 clusters, mean points of 8.70, and a density of  $0.0000655 \text{ per m}^2$ . VM has 111 clusters followed by AVM with 112 outperforming the other geomasks. The closest mean point value was obtained by DM at 8.66 as well as for the mean cluster density at  $0.0000598 \text{ per m}^2$ .

In Saxony 200, four clusters were generated by the original data with mean points at 6.5 and a mean cluster density of zero. DM succeeded the same values as original data whereas VM indicates the most different values. For Saxony 2000, the original data demonstrates 66 clusters, mean points at 7.86, and a mean cluster density at  $0.0000042 \text{ per m}^2$ . Regarding the first parameter, AVM outperforms the other methods at 63 clusters. Concerning the second parameter, DM reaches the closest mean points at 7.80. Finally, the most alike cluster density was obtained by AVM at  $0.0000044 \text{ per m}^2$ . AAE fares the worst with regard to the number of clusters and the mean cluster density while VM demonstrates the least efficiency for mean points.

Last, we derived the geomasks' ranks for each area data set/ metric and then the mode rank based on the divergence value (the closer to the original data value the higher is the rank). For both the mean points per cluster and mean cluster density, DM is ranked 1<sup>st</sup>, followed by AVM (2<sup>nd</sup>), then VM (3<sup>rd</sup>), and last is AAE (4<sup>th</sup>). For the cluster density, both DM and AVM are ranked as 1<sup>st</sup>, followed by AAE (2<sup>nd</sup>), then VM (3<sup>rd</sup>).

#### 4.5 Evaluation and comparison of geomasks

In the ESDA results subsections, we stated the ranking of each geomask. The final ranks are shown in table 4 to indicate the performance regarding data utility. AVM is ranked first for not displacing points to illogical locations or other residencies while VM is ranked first for retaining the extend of the original data. Hence, both geomasks are ranked as first for the visualization ESDA method because there are only these two metrics. The same applies to the central tendency (two metrics: mean and median), while for the nearest neighbour hierarchical cluster analysis we calculated the mode of the three metrics. DM is clearly the geomask that retains the pattern of the masked data the closest to the original one, while AAE distorts the pattern the most. Our proposed AVM method performs also very well and it is ranked as second regarding data utility.

Apart from data utility, this paper discussed the importance of preserving a level of SKA for the derived masked data. Unfortunately, trying to anonymize data sufficiently will eventually decrease their data utility. Also, displacing points to other domiciles should be avoided to prevent false re-identification. Hence, the optimal masking solution is to find the golden mean between these three aspects. These aspects are summarized in table 5, and compared across the geomasks. As stated before, the only method that prevents false re-identification is AVM. DM offers the best data utility, however, it only partially preserves a certain level of SKA because it assumes that the underlying population is homogeneously distributed. Both AVM and AAE retain a certain level of SKA while VM performs the

## 1:14 Adaptive Voronoi Masking (AVM)

worst considering all three aspects. By comparison, AVM is the optimal solution because it prevents false re-identification, offers a certain level of SKA, and is ranked second in terms of data utility.

■ **Table 3** Nearest neighbour hierarchical cluster analysis results for each geomask and area data set. The specific metrics are the number of clusters (number), mean points per cluster (points), and mean cluster density (density). Values closer to the original data values are marked in bold. DM has the closest values followed by the AVM.

Geomasks /Metric		Area data set					
		Leipzig 200	Leipzig 2000	Zwickau 200	Zwickau 2000	Saxony 200	Saxony 2000
Original data	Number	6	136	8	110	4	66
	Points	7.33	7.73	7.75	8.7	6.5	7.86
	Density	0.000013	0.0001144	0.0000055	0.0000655	0	0.0000042
AVM	Number	<b>7</b>	140	9	112	5	<b>63</b>
	Points	<b>7.5</b>	8.08	7.44	8.98	6.4	7.68
	Density	<b>0.0000113</b>	0.0008718	0.0000177	0.0004652	<b>0</b>	<b>0.0000044</b>
VM	Number	9	<b>139</b>	11	<b>111</b>	6	71
	Points	6.67	8.02	<b>7.82</b>	8.94	6.7	8.07
	Density	0.0000221	0.0001964	0.0000077	0.00010002	0.0000002	0.0000055
AAE	Number	<b>7</b>	57	5	44	3	36
	Points	6.43	7.02	6.6	8.25	7	7.58
	Density	0.0000169	0.0000835	0.0000028	0.0000511	<b>0</b>	0.000003
DM	Number	<b>7</b>	109	<b>8</b>	89	<b>4</b>	61
	Points	6.86	<b>7.83</b>	7.63	<b>8.66</b>	<b>6.5</b>	<b>7.8</b>
	Density	0.0000207	<b>0.0000979</b>	<b>0.000006</b>	<b>0.0000598</b>	<b>0</b>	0.0000037

■ **Table 4** Ranking of geomasking techniques based on their performance on four ESDA methods (visualisation of point pattern, central tendency, Ripley's K-function, and nearest neighbour hierarchical cluster analysis). DM retains the masked data pattern the most similar to the original data, followed by AVM.

Evaluation Method	Geomask (rank)			
	AVM	VM	AAE	DM
Visualization	1st	1st	4th	3rd
Central tendency	3rd	1st	4th	1st
Ripley's K-function	2nd	3rd	4th	1st
Nearest neighbour hierarchical cluster analysis	2nd	3rd	4th	1st
Mode Rank	<b>2nd</b>	<b>3rd</b>	<b>4th</b>	<b>1st</b>

■ **Table 5** Evaluation of geomasking techniques based on the ability to: a) prevent the risk of false re-identification, b) to ensure spatial K-anonymity, and c) to preserve original point pattern (data utility ranking). AVM offers the best combination of these three aspects (marked in bold).

Geomasks	False re-identification	Spatial K-anonymity	Data Utility
AAE	yes	yes	4th
<b>AVM</b>	<b>no</b>	<b>yes</b>	<b>2nd</b>
DM	yes	partly	1st
VM	yes	no	3rd

## 5 Conclusion

This study presented a new geographical masking method. AVM (i) considers the underlying population density by defining a level of K-anonymity, as AAE does, (ii) displaces a part of the original data based on the concept of VM, and (iii) by considering the underlying geography transfers points to the closest street intersection. Thus, it decreases the risk of false re-identification immensely and does not relocate data points to illogical positions.

The statistical analyses evidenced that AVM did not perform as well as DM regarding data utility, yet it was ranked as second among the four examined geomasks. Adding to that, it preserves the SKA accurately (AAE does this as well) and is the only method that does not dislocate points to illogical locations and minimizes the risk of false re-identification. However, it can be argued that a map viewer will view fewer data points (due to the street intersection aggregation) influencing the spatial perception of a phenomenon. Contrary to that, DM and VM, as well as AAE, can transfer data points to other residences or parcels increasing the risk of false re-identification. Based on three key factors (spatial K-anonymity, false re-identification, and data utility), it can be concluded that AVM is the most encouraging method in terms of the preservation of data utility and decreasing the risk of false re-identification to protect the individual's privacy.

Still, our method is not free of constraints (just like any geomasking method). For example, it might be a better approach to visualize a protected version of the distribution of a point pattern, but it will be less accurate in detecting local patterns compared to DM. Even more, it is a technique that can be successfully applied to confidential spatial data points but not to other geodata types. Location-enabled technologies capture geodata that are more complex and have to be treated/protected by different methods and privacy metrics [16, 20]. For instance, social media data capture, among other attributes, the spatiotemporal stamps of a user, which could be further processed to infer more than one type of spatial information (e.g., home or work locations). The evaluation of a method's efficiency regarding protection for this type of geodata should involve other measures and possibly be diversified by types of spatial information [11].

For the quality or information loss of masked data we applied four ESDA methods. Still, more methods can be implemented such as the global Moran's I for spatial autocorrelation or distance to K-nearest neighbour, as well as Local Indicators of Spatial Association. In addition, it is of great interest to examine the performance of AVM on national data sets. Furthermore, it is recommended to juxtapose AVM with more geomasks that were not applied here to gather more knowledge about the new approach.

Researchers and the public are becoming more aware of the privacy risks related to geodata. However, privacy guidelines as established by Kounadi and Resch [20] as well as the existence of geomasks have to become more well-known to researchers, institutions, companies, or the public sector. A first step to reach this goal is to make geomasks accessible and reproducible. During this research, it was discovered that only the geomask DM is retrievable online for free. This is confounding considering the fact that many researchers stress to mask confidential discrete spatial data. Our method is available for free via the Github repository "Geoprivacy"<sup>8</sup>. A further step is to employ geomasks for open-source software. Through that, companies, researchers, and institutions can share their data and findings with the public without jeopardizing individual privacy.

---

<sup>8</sup> <https://github.com/okounadi/Geoprivacy>

## References

- 1 Jayakrishnan Ajayakumar, Andrew J Curtis, and Jacqueline Curtis. Addressing the data guardian and geospatial scientist collaborator dilemma: how to share health records for spatial analysis while maintaining patient confidentiality. *International Journal of Health Geographics*, 18(1):1–12, 2019.
- 2 William B Allshouse, Molly K Fitch, Kristen H Hampton, Dionne C Gesink, Irene A Doherty, Peter A Leone, Marc L Serre, and William C Miller. Geomasking sensitive health data and privacy protection: an evaluation using an e911 database. *Geocarto international*, 25(6):443–452, 2010.
- 3 Marc P Armstrong, Gerard Rushton, and Dale L Zimmerman. Geographically masking health data to preserve confidentiality. *Statistics in medicine*, 18(5):497–525, 1999.
- 4 John S Brownstein, Christopher A Cassa, and Kenneth D Mandl. No place to hide—reverse identification of patients from published maps. *New England Journal of Medicine*, 355(16):1741–1742, 2006.
- 5 Christopher A Cassa, Shaun J Grannis, J Marc Overhage, and Kenneth D Mandl. A context-sensitive approach to anonymizing spatial surveillance data: impact on outbreak detection. *Journal of the American Medical Informatics Association*, 13(2):160–165, 2006.
- 6 Spencer Chainey, Lisa Tompson, and Sebastian Uhlig. The utility of hotspot mapping for predicting spatial patterns of crime. *Security journal*, 21(1-2):4–28, 2008.
- 7 National Research Council et al. *Putting people on the map: Protecting confidentiality with linked social-spatial data*. National Academies Press, 2007.
- 8 Philip M Dixon. R iple’s k function. *Wiley StatsRef: Statistics Reference Online*, 2014.
- 9 Matt Duckham and Lars Kulik. Location privacy and location-aware computing. *Dynamic & mobile GIS: investigating change in space and time*, 3:35–51, 2006.
- 10 Weijung J Fu, Peikun K Jiang, Guomo M Zhou, and Keli L Zhao. Using moran’s i and gis to study the spatial pattern of forest litter carbon density in a subtropical region of southeastern china. *Biogeosciences*, 11(8):2401, 2014.
- 11 Song Gao, Jimmeng Rao, Xinyi Liu, Yuhao Kang, Qunying Huang, and Joseph App. Exploring the effectiveness of geomasking techniques for protecting the geoprivacy of twitter users. *Journal of Spatial Information Science*, 2019(19):105–129, 2019.
- 12 Christopher Graham. Anonymisation: managing data protection risk code of practice. *Information Commissioner’s Office*, 2012.
- 13 Ruchika Gupta and Udai Pratap Rao. Preserving location privacy using three layer rdv masking in geocoded published discrete point data. *World Wide Web*, 23(1):175–206, 2020.
- 14 Danielle F Haley, Stephen A Matthews, Hannah LF Cooper, Regine Haardörfer, Adaora A Adimora, Gina M Wingood, and Michael R Kramer. Confidentiality considerations for use of social-spatial data on the social determinants of health: Sexual and reproductive health case study. *Social Science & Medicine*, 166:49–56, 2016.
- 15 Kristen H Hampton, Molly K Fitch, William B Allshouse, Irene A Doherty, Dionne C Gesink, Peter A Leone, Marc L Serre, and William C Miller. Mapping health data: improved privacy protection with donut method geomasking. *American journal of epidemiology*, 172(9):1062–1069, 2010.
- 16 Carsten Keßler and Grant McKenzie. A geoprivacy manifesto. *Transactions in GIS*, 22(1):3–19, 2018.
- 17 Ourania Kounadi and Michael Leitner. Why does geoprivacy matter? the scientific publication of confidential data presented on maps. *Journal of Empirical Research on Human Research Ethics*, 9(4):34–45, 2014.
- 18 Ourania Kounadi and Michael Leitner. Spatial information divergence: Using global and local indices to compare geographical masks applied to crime data. *Transactions in GIS*, 19(5):737–757, 2015.

- 19 Ourania Kounadi and Michael Leitner. Adaptive areal elimination (aae): A transparent way of disclosing protected spatial datasets. *Computers, Environment and Urban Systems*, 57:59–67, 2016.
- 20 Ourania Kounadi and Bernd Resch. A geoprivacy by design guideline for research campaigns that use participatory sensing data. *Journal of Empirical Research on Human Research Ethics*, 13(3):203–222, 2018.
- 21 Mei-Po Kwan, Irene Casas, and Ben Schmitz. Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks? *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(2):15–28, 2004.
- 22 Michael Leitner and Andrew Curtis. Cartographic guidelines for geographically masking the locations of confidential point data. *Cartographic Perspectives*, (49):22–39, 2004.
- 23 Gerard Rushton, Marc P Armstrong, Josephine Gittler, Barry R Greene, Claire E Pavlik, Michele M West, and Dale L Zimmerman. *Geocoding health data: the use of geographic codes in cancer prevention and control, research and practice*. CRC Press, 2007.
- 24 Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.
- 25 Bill Schilit, Jason Hong, and Marco Gruteser. Wireless location privacy protection. *Computer*, 36(12):135–137, 2003.
- 26 Klaus Schwab, Alan Marcus, JO Oyola, William Hoffman, and Michele Luzi. Personal data: The emergence of a new asset class. In *An Initiative of the World Economic Forum*, 2011.
- 27 Dara E Seidl, Piotr Jankowski, and Keith C Clarke. Privacy and false identification risk in geomasking techniques. *Geographical Analysis*, 50(3):280–297, 2018.
- 28 Dara E Seidl, Piotr Jankowski, and Atsushi Nara. An empirical test of household identification risk in geomasked maps. *Cartography and Geographic Information Science*, 46(6):475–488, 2019.
- 29 Dara E Seidl, Gernot Paulus, Piotr Jankowski, and Melanie Regenfelder. Spatial obfuscation methods for privacy protection of household-level data. *Applied Geography*, 63:253–263, 2015.
- 30 Paul A Zandbergen. Ensuring confidentiality of geocoded health data: assessing geographic masking strategies for individual-level data. *Advances in medicine*, 2014, 2014.
- 31 Su Zhang, Scott M Freundsuh, Kate Lenzer, and Paul A Zandbergen. The location swapping method for geomasking. *Cartography and Geographic Information Science*, 44(1):22–34, 2017.



# Reproducible Research and GIScience: An Evaluation Using GIScience Conference Papers

Frank O. Ostermann<sup>1</sup>  

Faculty of Geo-Information Science and Earth Observation (ITC),  
University of Twente, Enschede, The Netherlands

Daniel Nüst  

Institute for Geoinformatics, University of Münster, Germany

Carlos Granell  

Institute of New Imaging Technologies, Universitat Jaume I de Castellón, Spain

Barbara Hofer  

Christian Doppler Laboratory GEOHUM and Department of Geoinformatics - Z\_GIS,  
University of Salzburg, Austria

Markus Konkol  

Faculty of Geo-Information Science and Earth Observation (ITC),  
University of Twente, Enschede, The Netherlands

---

## Abstract

GIScience conference authors and researchers face the same computational reproducibility challenges as authors and researchers from other disciplines who use computers to analyse data. Here, to assess the reproducibility of GIScience research, we apply a rubric for assessing the reproducibility of 75 conference papers published at the GIScience conference series in the years 2012-2018. Since the rubric and process were previously applied to the publications of the AGILE conference series, this paper itself is an attempt to replicate that analysis, however going beyond the previous work by evaluating and discussing proposed measures to improve reproducibility in the specific context of the GIScience conference series. The results of the GIScience paper assessment are in line with previous findings: although descriptions of workflows and the inclusion of the data and software suffice to explain the presented work, in most published papers they do not allow a third party to reproduce the results and findings with a reasonable effort. We summarise and adapt previous recommendations for improving this situation and propose the GIScience community to start a broad discussion on the reusability, quality, and openness of its research. Further, we critically reflect on the process of assessing paper reproducibility, and provide suggestions for improving future assessments. The code and data for this article are published at <https://doi.org/10.5281/zenodo.4032875>.

**2012 ACM Subject Classification** Information systems → Geographic information systems

**Keywords and phrases** reproducible research, open science, reproducibility, GIScience

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.2

**Related Version** *Previous Version:* <https://doi.org/10.31223/X5ZK5V>

**Supplementary Material** The input data for this work are the full texts of GIScience conference proceedings from the years 2012 to 2018 [35, 7, 20, 34]. The paper assessment results and source code of figures are published at <https://github.com/nuest/reproducible-research-at-giscience> and archived on Zenodo [27]. The used computing environment is containerised with Docker pinning the R version to 3.6.3 and R packages to the MRAN snapshot of July 5th 2019.

**Funding** *Daniel Nüst:* Project *o2r*, German Research Foundation, grant number PE 1632/17-1.

*Carlos Granell:* Ramón y Cajal Programme of the Spanish government, grant number RYC-2014-16913.

*Markus Konkol:* Project *o2r*, German Research Foundation, grant numbers KR 3930/8-1 and TR 864/12-1.

---

<sup>1</sup> Corresponding author



**Acknowledgements** Author contributions (see CRediT): *all authors* contributed to conceptualisation, investigation (number of assessed papers in brackets), and writing – original draft; FO (33): writing - review & editing, software; DN (33): software, writing - review & editing, visualisation; CG (30): writing - review & editing, software; BH (21): writing - review & editing; MK (30). We thank Celeste R. Brenneka from the Scientific Editing Service of the University of Münster for her editorial support and the anonymous reviewers for their constructive feedback.

## 1 Introduction

The past two decades have seen the imperative of Open Science gain momentum across scientific disciplines. The adoption of Open Science practices is partially prompted by the increasing costs of using proprietary software and subscribing to scientific journals, but more importantly because of the increased transparency and availability of data, methods, and results, which enable reproducibility [22]. This advantage is especially relevant for the computational and natural sciences, where sharing data and code is a prerequisite for reuse and collaboration. A large proportion of GIScience research today uses software to analyse data on computers, meaning that many articles published in the context of the GIScience conference series<sup>2</sup> fall into the categories of data science or computational research. Thereby, these articles face challenges of transparency and reproducibility in the sense of the Claerbout/Donoho/Peng terminology [2], where *reproduction* means a recreation of the same results using the same input data and methods, usually with the actual code created by the original authors. The related concept of replication, i.e., the confirmation of insights gained from a scientific study using the same method with new data, is of crucial importance to scientific progress, yet it is also frequently challenging to realise for interested readers of a published study. So far, despite the GIScience conference series' rigorous review process, reproducibility and replicability have not been a core concern in the contributions. With reproducibility now being a recognised topic in the call for papers, it is time to take stock and identify possible action. In previous work [26], we assessed the reproducibility of a selection of full and short papers from the AGILE conference series<sup>3</sup>, a community conference organised by member labs of the Association of Geographic Information Laboratories in Europe (AGILE). Using systematic analysis based on a rubric for reproducible research, we found that the majority of AGILE papers neither provided sufficient information for a reviewer to evaluate the code and data and attempt a reproduction, nor enough material for readers to reuse or extend data or code from the analytical workflows. This is corroborated by research in related disciplines such as quantitative geography [3], qualitative GIS [21], geoscience [16], and e-Science [10]. The problems identified in these related research areas are transferable to the scientific discipline of GIScience, which operates at the intersections of aforementioned fields [11]. In any case, observations on the lack of reproducibility in all scientific fields contrast with the clear advantages and benefits of open and reproducible research both for individuals and for academia as a whole (cf. for example [6, 19, 17, 5]). As a consequence, we have initiated a process to support authors in increasing reproducibility for AGILE publications; as a main outcome, this initiative has produced author guidelines as well as strategies for the AGILE conference series<sup>4</sup>.

---

<sup>2</sup> <https://www.giscience.org/>

<sup>3</sup> <https://agile-online.org/conference>

<sup>4</sup> See the initiative website at <https://reproducible-agile.github.io/>, the author guidelines at <https://doi.org/10.17605/OSF.IO/CB7Z8> [24] and the main OSF project with all materials <https://osf.io/phmce/> [25].

The AGILE conference is related to GIScience conference in terms of scientific domain and contributing authors, but is different in organisational aspects. Two open questions are thus whether the GIScience conference series faces the same issues, and whether similar strategies could be applied successfully. To begin this investigation, we conducted a simple text analysis of GIScience conference proceedings<sup>5</sup> to evaluate the relevance of computational methods in the conference papers. The analysis searched for several word stems related to reproducibility: Generic words indicating a quantitative analysis, e.g., “data”, “software”, or “process”; specific platforms, e.g., “GitHub”; and concrete terms, e.g., words starting with “reproduc” or “replic”. Table 1 shows the results of the search for each year analysed. The take-away message from the text analysis is that algorithms, processing, and data play an essential role in GIScience publications, but few papers mentioned code repositories or reproduction materials. Therefore, an in-depth assessment of the reproducibility of these publications was deemed necessary.

The main contribution of this work addresses two objectives: First, it aims to investigate the state of reproducibility in the GIScience conference community. This investigation broadens our knowledge base about reproducibility in the GIScience discipline and informs us about the situation in the GIScience conference series specifically (details in section 4). Second, it aims to apply the assessment procedure used for AGILE conference papers (presented in section 3) to the papers of the GIScience conference, so that the broader suitability of this procedure is evaluated using a different dataset, and thereby providing evidence of its replicability. Such a transfer validates the developed methodology. We discuss these findings and present our conclusions in the final two sections (5 and 6). Together, these objectives yield important findings for the discussion of reproducibility within the GIScience conference community and the GIScience discipline at large. We believe that GIScience as a discipline would greatly benefit from more studies that reproduce and replicate other studies, similar to other disciplines that are recognising the value of replication for innovating theory [23], and argue that such a replication study is not lacking innovation but is a prerequisite for innovating community practice. Only then can a fruitful dialogue take place on whether and how to improve reproducibility for the GIScience conference series, and whether the recent steps taken at AGILE<sup>6</sup> could be an inspiration for GIScience conferences as well.

## 2 Related work

This work builds and expands on earlier work [26], which already provides an overview of reproducible research in general, including definitions, challenges, and shortcomings. In the following, we focus therefore on recently published works and briefly introduce related meta-studies.

Few groups have attempted practical reproduction of computational works related to GIScience. Konkol et al. [16] conducted an in-depth examination of the computational reproducibility of 41 geoscience papers with a focus on differences between the recreated figures. The set of papers was, similar to our work, drawn from a fixed group of two outlets

---

<sup>5</sup> The full text analysis and the results is available in this paper’s repository in the following files: `gis-science-historic-text-analysis.Rmd` contains the analysis code; the result data are two tables with counts for occurrences of words respectively word stems per year in `results/text_analysis_topwordstems.csv` and `results/text_analysis_keywordstems.csv`; a word-cloud per year is in file `results/text_analysis_wordstemclouds.png`.

<sup>6</sup> See the initiative website at <https://reproducible-agile.github.io/>, the author guidelines at <https://doi.org/10.17605/OSF.IO/CB7Z8> [24] and the main OSF project with all materials <https://osf.io/phmce/> [25].

■ **Table 1** Reproducibility-related word stems in the corpus per year of proceedings.

year	words	reproduc...	replc...	repeatab...	code	software	algorithm(s)	(pre)process..	data.*	result(s)	repository/ies	github/lab
2002	23782	6	2	0	11	61	191	150	897	129	62	0
2004	26728	4	1	0	34	50	138	258	849	263	4	0
2006	32758	6	0	0	12	32	335	250	856	164	0	0
2008	27356	3	6	1	3	11	331	146	854	218	17	0
2010	23004	3	1	0	8	16	164	276	650	162	0	0
2012	28860	2	0	0	101	27	238	190	1048	311	3	0
2014	29534	3	4	1	12	18	255	159	1070	228	3	0
2016	24838	2	0	0	23	21	333	150	1007	202	4	1
2018	23318	3	10	0	15	15	201	160	891	294	6	6
Total	240178	32	24	2	219	251	2186	1739	8122	1971	99	7

*Note:* The very high value for 'code' in 2012 is due to a single paper about land use, for which different "land use codes" are defined, discussed and used.

(journals), but it was further limited to recent papers providing code in the R language. The main issues raised by Konkol et al. [16] are similar to those identified in a recent report on the reproducibility review during the AGILE conference 2020<sup>7</sup>, where the reproducibility committee summarised the process and documented relevant obstacles to reproducibility of accepted papers.

Within the geospatial domain, Kedron et al. [13] provide a recent review of opportunities and challenges for reproducibility and replicability. They transfer solutions from other domains but also discuss and conceptualise the specific nature of a reproducibility and replicability framework when working with geospatial data, e.g., handling context, uncertainty of spatial processes, or how to accommodate the inherent natural variability of geospatial systems. In a similar manner, Brunson and Comber [4] investigate reproducibility within spatial data science, with special attention to big spatial data. They support the need for open tools, knowledge about code, and reproducibility editors at domain journals and conferences, but they also introduce the perspective that spatial analysis is no longer conducted only by GI/geo-scientists or geographers and connect reproducibility with critical spatial understanding. The more conceptual work in those articles is complemented by the assessment of reproducibility conducted in this paper.

Two recent studies from distant disciplines, wildlife science [1] and hydrology [32], also relate to our work in this paper. Both studies investigate a random set of articles from selected journals and use a stepwise process of questions to determine the availability of materials and eventually reproduce workflows if possible. Archmiller et al. [1] use a final ranking of 1 to 5 to specify the degree to which a study's conclusions were eventually reproduced. Similar to our classification scheme, their ranking models fit the general notion of a "reproducibility spectrum" [30].

<sup>7</sup> <https://osf.io/7rjpe/>

## 3 Reproducibility assessment method

### 3.1 Criteria

The assessment criteria used for the current study were originally defined in previous work, so we provide only a short introduction here and refer to Nüst et al. [26] for details. The three assessment criteria are *Input Data*, *Methods*, and *Results*. *Input Data* comprises all datasets that the computational analysis uses. *Methods* encompasses the entire computational analysis that generates the results. Since *Methods* is difficult to evaluate as a whole, we split this criterion into three subcriteria: *Preprocessing* includes the steps to prepare the *Input Data* before the main analysis; *Methods, Analysis, Processing* is the main analysis; *Computational Environment* addresses the description of hard- and software. Finally, the criterion *Results* refers to the output of analysis, e.g., figures, tables, and numbers.

For each of these (sub)criteria, we assigned one of four levels unless the criterion was not applicable (*NA*). *Unavailable* (level 0) means that it was not possible to access the paper's data, methods, or results, and that it was impossible to recreate them based on the description in the paper. *Documented* (level 1) indicates that the paper still did not provide direct access to datasets, methods, or results, but that there was sufficient description or metadata to potentially recreate them closely enough for an evaluation; yet, often a recreation was unlikely due to the huge amount of effort needed. For example, with regard to the methods criteria, *Documented* means that pseudo code or a textual workflow description was provided. *Available* (level 2) was assigned if the paper provided direct access to the materials (e.g., through a link to a personal or institutional website), but not in the form of an open and permanent identifier, such as a digital object identifier (DOI). The indication of a DOI does not apply to the methods criteria, as it is not yet common practice to make a permanent reference to code, libraries, and system environments with a single identifier. The gold standard, *Available and Open* (level 3), requires open and permanent access to the materials (e.g., through public online repositories) and open licenses to allow use and extension.

Note that levels are ordinal numbers that can be compared (3 is higher than 2), but absolute differences between numbers must not be interpreted as equals: Moving one level up from 0 to 1 is not the same as from level 1 to level 2. While reaching level 1 is fairly straightforward, moving to level 2 means one must create a fully reproducible paper.

### 3.2 Process

The overall approach to assessing the reproducibility of GIScience papers followed the previous assessment of AGILE papers [26], and was conducted by the same persons. Contrary to the AGILE investigation, all full papers in the GIScience conference series (from the 2012 to 2018 editions) were assessed. This is partly because no obvious subset exists, such as the nominees for best papers as in the case of the AGILE conference series, but also because we aimed to work with a larger dataset for potentially more informative results. Each GIScience conference paper was randomly assigned to two assessors who evaluated it qualitatively according to the reproducibility criteria. The assessors were free in the way they approached the assigned evaluations, depending on the structure of the paper and the assessor's familiarity with the topic. An evaluation could range from browsing the paper to identify relevant statements in case of high familiarity to a thorough reading of the full text. The identification of relevant content could be supported to some extent by a PDF reader with multiple highlights, using keywords like e.g., "data, software, code, download,

contribution, script, workflow”. The results of the individual assessments were joined in a collaborative Google Spreadsheet. This spreadsheet also had a comments column for assessors to record relevant sources and decisions. In case of disagreement between assessors, arguments for and against a certain reproducibility level were discussed in the entire group of five assessors until a consensus was reached. Only then were the assessments merged into a single value. A snapshot of both the unmerged and merged values was stored as a CSV file in the collaboration repository for transparency and provenance<sup>8</sup>. Two independent assessors per paper increased the objectivity of the final assessment. Disagreements and conducting the assessment one year at a time, going backwards from the most recent year, were found helpful in aligning the interpretation of criteria and, in rare cases, led to an adjustment of similar cases in other papers.

The discussion about the correct assignment of levels led to a reflection on how to apply the rubric for special situations. For the *Input Data* criterion, some papers had input data “available” at the time of writing/publication that was not available anymore at the time of evaluation, due to broken links, changes in the URL structure of a website, or projects and/or personal websites that were down or moved. In such cases, we gave the authors the benefit of the doubt and assumed the data were accessible some time after the publication of the conference proceedings. We did not give those papers an arbitrary score and discussed internally the best level per case; yet, such papers never earned a 3, which would require permanent resolving of the link. Related to this criterion, simulation data, like the specification or configuration of agents in an agent-based system, was not treated as input data (resulting in *NA* if no other data was used), but as parameters of the main analysis, i.e., as part of the *Methods, Analysis, Processing*.

*Preprocessing* covers preparatory work for the actual analysis involving various tasks such as data selection, cleaning, aggregation, and integration. However, the dividing line between data preprocessing and processing (i.e., the main analysis) proved to be often vague, and occasionally assessors disagreed whether the preprocessing criterion should be assigned *NA*, *Unavailable*, or *Documented* (0 or 1, respectively). Therefore, we decided eventually to apply the *Preprocessing* criterion only in cases where papers specifically mentioned a preprocessing task independent of the actual analysis or method, e.g., when clearly stated in a separate sub-section of the paper.

Lastly, human subject tests and surveys were also a special case. Human-related research activities were rated as 1 in the methods/analysis/processing criterion if sufficiently documented; nonetheless, a sufficient documentation in these cases did not mean that original sources were available or could be exactly recreated.

### 3.3 Paper corpus

In total, 87 papers from the GIScience conferences in 2012, 2014, 2016, and 2018 were assessed. A table in the reproducibility package shows the full results of the assessment and the included raw data provides details on assigned assessors, authors, etc. [27]. 12 papers (14%) across all years were identified as conceptual papers<sup>9</sup> and were not included in the corpus. The number of conceptual papers in GIScience conferences was low over the analysed

---

<sup>8</sup> The assessment results are in the file `results/paper_assessment.csv`. As an example, commit `464e630` and `2e8b1be` are the pre-merge and post-merge commit after completing the assessment of the papers from 2014. The pre-merge commit contains the assessments including the assessors’ initials, e.g. “CG: 1, MK: 1”.

<sup>9</sup> See [26] for a definition of “conceptual”.

■ **Table 2** Statistics of reproducibility levels per criterion (rounded to one decimal place).

	input data	preproc.	method/analysis/proc.	comp. env.	results
Min.	0.0	0.0	0	0.0	0.0
Median	1.0	1.0	1	0.0	1.0
Mean	0.7	0.8	1	0.3	1.1
Max.	2.0	2.0	2	1.0	2.0
NA's	1.0	24.0	0	0.0	0.0

years (2012: 4; 2014: 5; 2016: 3), and none in 2018. This might suggest an increasingly predominant and ubiquitous role of analytical datasets and computational workflows in the generation of the final published results in the field.

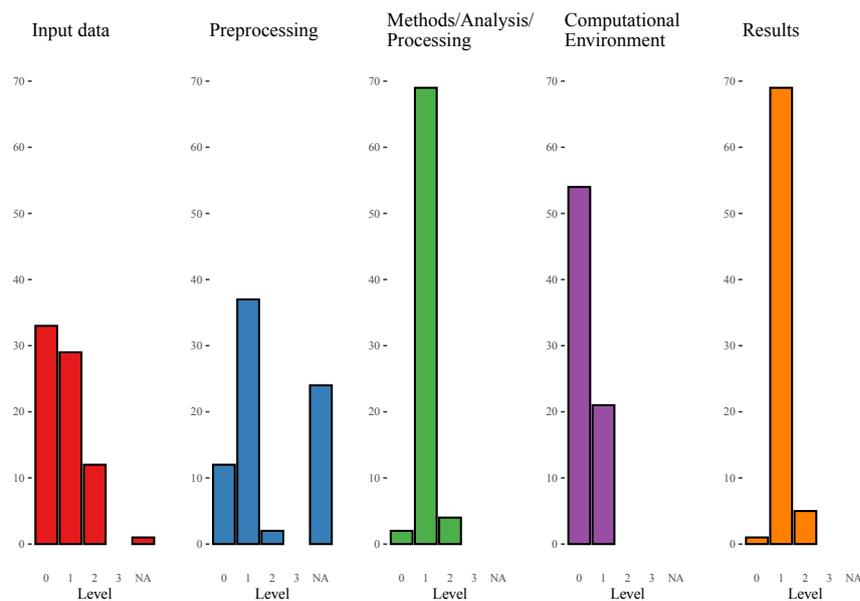
#### 4 Reproducibility of GIScience conference papers

Table 2 shows aggregated values for the assessed reproducibility levels. If we look at the median values of the five criteria (Table 2), a typical GIScience paper scores 1 1 1 0 1. This score translates in practical terms into a paper that is sufficiently documented to claim that reproduction could be attempted within a reasonable time frame after publication. While such a level of reproducibility is typically accepted by journals and conferences today, it does not guarantee that a reproduction would be possible and practical. A reproduction of such a paper would require considerable effort, namely technical skills, communication with authors, and time not only to both gather, recreate, and/or analyse all the necessary resources (data, code, etc.) but also to recreate the specific computational environment of the paper. Especially the latter is very unlikely, as the computational environment is generally not specified at all, as demonstrated by the median value of 0 (*Unavailable*) for this sub-criterion.

Figure 1 shows the distribution of the reproducibility levels for each criterion. None of the papers reached the highest reproducibility level of 3 (*Available and Open*) on any criterion. Only 12 papers reached level 2 (*Available*) in the *Input Data* criterion. Similar to previous results [26], the number of papers with level 0 for *Input Data* was especially high (33, corresponding to 44%), which is a significant barrier to reproduction since input data is not only unavailable but also cannot be recreated from the information provided in the paper.

*Preprocessing* applied to only 51 publications. For 24 papers, the *Preprocessing* criterion was not applicable (*NA*). This large number is a result of our decision to assess *Preprocessing* only if papers explicitly stated or described a preprocessing step in their analysis, which few did. This does not mean the assessment ignored missing information on preprocessing step, only that such missing information would then reduce the level of the *Methods* criterion instead. Obviously, if data preprocessing is required but it is either not indicated in the paper or is not provided as an additional (computational) step or resource, the ability to reproduce the paper will be limited. The achieved levels for *Preprocessing* remained low: 37 papers reach level 1 (*Documented*), about half of the papers with level 1 in the *Methods* criterion. For the other half, it was not clear whether data preprocessing tasks existed at all, or whether these tasks were part of the main analysis.

*Methods* and *Results* criteria show a similar distribution (see Figure 1). Indeed, 65 publications had level 1 in both criteria, which represents 87% of the papers assessed. In this sense, most of the assessed papers fall below the minimum standard for reproduction in the methods and results criteria. All papers except one reached level 1 for the *Results* criterion, which shows that the peer review worked as expected for almost all articles. In other words,



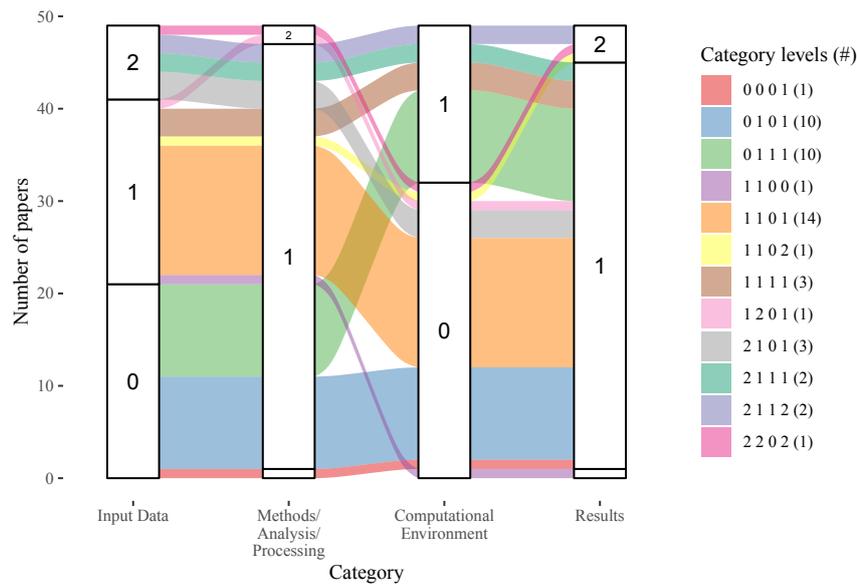
■ **Figure 1** Barplots of reproducibility assessment results; levels range from 0 (leftmost bar) to 'not applicable' (rightmost bar).

authors are concerned with making the results understandable to the reviewers, which is not always the case for the other criteria. More generally, this aspect raises the question of whether peer review should stop in the absence of minimal evidence of the input data, analysis, and computational environment used in a paper.

Finally, papers scored worse on the *Computational Environment* criterion. Overall, 54 publications (72%) remained at level 0, which means that no information was provided in the paper about the computing environment, tools, or libraries used in the reported analysis. The *Computational Environment* criterion and the *Input Data* criterion accounted for a significant number of 0 values, which clearly signals an impediment to reproduction. It also shows a rather low recognition of data and software as academic outputs, because both data and software should be properly cited to give credit to their creators [18, 12].

Figure 2 shows an alluvial diagram of all scores, i.e., combinations of criteria values of those 49 papers without any *NA* criterion. Most of the excluded papers have *NA* for *Preprocessing*, therefore this criterion is not included in the figure. The diagram confirms overall patterns seen before. The vast majority of papers have level 1 in *Methods/Analysis/Processing* and *Results*. *Input data* is most diverse, with a surprisingly large number of papers with level 0 but also the largest fraction of papers reaching level 2. Many papers show low levels in *Computational Environment*.

The diagram illustrates how groups of papers with similar properties “flow” through the different criteria. Three major groups, which represent 34 of the papers (69%) included in the figure, become visible as broad bands. Two groups with 10 papers each start with level 0 for *Input Data* and 1 for *Methods/Analysis/Processing* and reach a 1 for *Results*, while they are divided equally between level 0 and 1 for *Computational Environment*. These two groups seem to indicate that the authors and reviewers alike follow the established pattern that results outweigh concerns for transparency and reproducibility, since computational papers with *Unavailable* input data are irreproducible. The third and largest group matches the overall mean values for the typical GIScience paper with level 1 for all criteria except for *Computational Environment*.



■ **Figure 2** Alluvial diagram of common groups of papers throughout 4 of 5 categories including only papers without any “not applicable” (*Level NA*) value; category *Preprocessing* was dropped because difficulty to clearly assess it lead to many “not applicable” values.

The diagram also shows additional interesting patterns for a few papers. The papers with the lowest level of 0 in *Results*, i.e., according to the assessors the results are documented insufficiently and thus difficult or impossible to fully understand, actually have better values in previous criteria. Only few papers that start with level 2 in *Input Data* can keep this level for *Methods/Analysis/Processing*, and even those who do later drop to level 0 in *Computational Environment*. Only one paper each shows the following surprising paths: Starting with level 1 for *Input Data*, then moving up to level 2 in *Methods*, before reaching level 2 in *Results* despite having only values of 1 or 0 in other criteria. In summary, not a single paper can reach the required levels for an immediate reproduction by ensuring that all required pieces are *Available* (level 2), not even considering the further challenges for reproductions, such as incomplete documentation [28]. An investigation of yearly scores to track developments over time does not show any trend, i.e., there is little change in reproducibility over the study period<sup>10</sup>. The overall low values for *Computational Environment* are one signal that confirms the growing concerns for reproducibility and reusability of computational research are not misplaced.

## 5 Discussion

### 5.1 State of reproducibility in the GIScience conference series

Our first research objective was to assess the state of reproducibility in the GIScience conference series. A recurrent issue found in the analysis was the inability to access input data based on the information provided in the paper. Most of the links and pointers to

<sup>10</sup> See the additional analysis and plots published at <https://nuest.github.io/reproducible-research-at-giscience/giscience-reproducibility-assessment.html> or in the paper’s reproducibility package [27].

datasets reported at the time of publication were either broken (e.g., non-existing resource, HTTP 404 error, invalid URL syntax) or not available anymore (URL works but redirects to a different generic page; specific resource from the paper no longer exists). In these cases, a level 2 in the *Input Data* criterion was deserved at the time of publication; however, when evaluating the level of reproducibility some time later, as was done in this work, level 2 is no longer suitable for those papers. From a reproducibility point of view, the input data was therefore not accessible, although contacting the authors could still be attempted. However, according to the meaning of the criterion and in practical terms, this is equivalent to including the statement “available upon request” in the paper and thereby level 0. An important part of reproducibility is that access to material should not degrade over time, which is best achieved by depositing data in repositories, including sensitive data (using the appropriate mechanisms), and properly citing it. In this assessment of reproducibility, we decided to give the authors the benefit of the doubt and awarded a value of 2 for *Input Data* even if we could not conclusively determine, e.g., by using the Internet Archive’s Wayback Machine<sup>11</sup>, whether the original website ever existed.

Regarding the common situation of a paper with *Documented* (level 1) for all criteria, our interpretation is that this is indeed a regular paper that is up to current scientific standards. Does this imply that a paper with *Unavailable* (level 0) in any criterion should not have been accepted? We believe that this requires differentiation between past and future papers. The criteria used in this paper were not included in the previous call for papers or in the reviewer guidelines, and therefore received less attention from authors or reviewers. Thus, we have analysed work in a historical context when there were few concrete incentives to push these aspects, beyond the general concerns for good scientific practice. Nowadays, with awareness about reproducibility being raised through initiatives, projects, and publications about it, we would expect that reproducibility levels increase, and argue that papers with *Unavailable* in one more criteria should not be accepted anymore without a clear and explicit justification (e.g., sensitive data on human subjects). This does not imply that it is always necessary to achieve the gold standard of *Available and Open*. The overall objective should be to make a paper as reproducible as possible before publication. We argue that, for most currently published works at the GIScience conference, *Available* would have been achievable and feasible with reasonable efforts.

However, such a change in standards for paper acceptance would also mean that researchers, editors, and publishers might have to reevaluate their focus on publishing novel and purportedly groundbreaking results in science, and give as much weight to publishing the full process and collection of parts that would allow readers to try to fully understand the research. Clearly, *Unavailable* for *Input Data* is the most problematic, because without sufficient knowledge about the characteristics of the input data, all attempts at reproducing results are bound to fail, even when the textual documentation of the data would potentially allow for an time-intensive recreation of the computational workflow.

## 5.2 Transferability of method

Concerning our second research objective, we can state that the overall process and the application of the reproducibility rubric was successfully replicated with a different data set. This is not entirely surprising given that AGILE and GIScience conference series share similarities in target audience, review process, and publication of proceedings (more on

---

<sup>11</sup><https://web.archive.org/>.

that in the following section). More importantly, the process faced similar challenges as we recalled from its earlier application. This is crucial information, because the successful replication of the process, including its challenges, enables us and others to ground any changes in solid evidence. In particular the *Preprocessing* criterion caused many discussions among the reproducibility reviewers during the assessment. It is often not clear or a matter of interpretation if a particular processing step belongs to a minor basic transformation of input data, if it is already part of the main analysis, and when it is a truly distinct step in the process. The borders are vague and hence scores should be interpreted with caution. Likewise, the *Computational environment* is also difficult to distinguish from analysis, and technology and practices for the effective management of the computing environment have reached mature states relatively recently. Future reproducibility assessments of papers could provide a more precise definition for *pre-processing*, e.g., only use it if the authors use the term, or might consider to drop the category, and benefit from rules to deal with the specific issues of older workflows, similar as discussed for input data above. Furthermore, it is important to remember that the levels of reproducibility are not equidistant in the sense that a level of 2 would be twice as good as a level of 1, or that the effort needed is twice as high. A level of 1 should be the standard for current and future peer-reviewed papers. Reaching level 2 requires several additional steps, while reaching the gold standard of 3 is again a comparatively small step from level 2 in terms of effort - the main difference is to use public repositories with a DOI - yet with a high positive impact on permanent accessibility.

Although the replication was successful, the process was again labour-intensive, making it problematic to scale it up to assess multiple years of several popular journals, for example. Further, despite our best efforts for transparency and the four-eyes principle in the assessment, the process is inherently subjective. A different group of investigators might score papers differently. While natural language processing techniques have made great progress in the past decades, an automated assessment of a paper's reproducibility still seems out-of-reach. Including important information as machine-readable metadata could allow to come closer to automation.

### 5.3 Comparison of conferences

Given that we followed the same process as in [26] and demonstrated the transferability of the method, comparing the two conference series seems appropriate. It is important to remember that we do not attempt such a comparison with the objective of declaring a “winner”. The published work and contributing community of the two conferences are similar enough for a comparison, yet their organisation (setup, process, geographic focus) differ too much for a simplistic ranking. However, a comparison is required to sensibly discuss whether the guidelines developed for AGILE might also be promising for GIScience: Are they transferable? If not, what adaptations seem necessary?

Concerning the contributing and participating academic communities, Egenhofer et al. [8] and Kemp et al. [14] both include both conferences series as outlets for GIScience research. Further, Keßler et al. [15] investigate the bibliographies of four GIScience conference series, including GIScience and AGILE for the year 2012, and identify 15 authors who have published in both conference series. We conducted a cursory investigation of the body of authors for full papers, revealing significant overlap<sup>12</sup>: Out of 571 unique AGILE and 405 unique GIScience full paper authors, 86 published in both conferences, and this includes all 15 authors mentioned by Keßler et al. [15]. Therefore, the strong relation between the AGILE and GIScience

---

<sup>12</sup>The data and code for the brief exploration into the authorship across the conferences considered in this work can be found in the directory `author_analysis` of this paper's reproducibility package [27].

■ **Table 3** Mean values per criterion for both conferences (rounded to two decimal places).

Criterion	AGILE full papers	GIScience papers
input data	0.67	0.72
method/analysis/processing	1.00	1.03
computational environment	0.62	0.28
results	0.88	1.05

conference series confirms our approach to apply the same methodology to GIScience that has been developed for AGILE conference publications, and it might lead to similar implications for improving reproducibility.

Nevertheless, before discussing any strategies to improve reproducibility, it is important to identify and consider the differences between the two conference series. GIScience is a biannual conference series whereas AGILE is annual, and they feature different pre-publication review processes and review management systems: In AGILE both authors and reviewers are anonymous, while in GIScience only the reviewers are. Furthermore, the AGILE conference series has the AGILE association<sup>13</sup> as an institutional supporter, which means a more stable organisational and financial framework for activities spanning more than one or between conferences. However, like GIScience, local conference organisers for AGILE have the main financial burden and experiences are informally handed over between organising committees. Geographic focus is also different: GIScience has a global target audience, and the individual conferences are likely to be different in their contributor communities because of the moving conference location, which often means lowered accessibility for authors from other parts of the world. AGILE, by comparison, has a European focus and accessibility is more homogeneous, although the conference location moves every year,. This likely translates into a less fluctuating and less geographically diverse audience at AGILE. Clearly, these observations will need a reassessment in several years to evaluate the impact of both conferences going full online in 2020/21 because of the travel and activity restrictions due to the COVID-19 pandemic.

Concerning the paper corpora, the publication years considered here (2012-2018) are similar to the assessment of AGILE papers (2010-2017), which makes the results comparable in the sense of what methods and tools would have been available for authors. Furthermore, we note that both conferences have a similar ratio of conceptual papers which were not assessed for reproducibility: In the AGILE corpus we identified 5 of 32 conceptual papers (15.6%), in the GIScience corpus there were 12 of 87 (13.8%). This indicates that both conferences have similar share of papers that used, at least in part, computational methods. On the content of the papers, our overall impression was that a larger share of GIScience papers included theoretical, conceptual, or methodological aspects, while AGILE papers seemed to feature more empirical and/or applied geoinformation science research.

Regarding the results of the reproducibility assessments as summarised in Table 3, the nature of the data and sample size does not support statistical analyses on significant differences. Nevertheless, looking at the *Input Data* criterion, GIScience has a slightly higher mean value compared to AGILE full papers (0.72 as opposed to 0.67) and a median of 1. These values indicate that the GIScience contributions had a slightly better, but by no means optimal, availability of input data. The pattern of reproducibility of the papers' workflows (category *Method, Analysis, Processing*) was very similar for the two conference

<sup>13</sup><https://agile-online.org/>.

series: The majority of papers achieved a level of 1, resulting in a mean of 1.03 for GIScience and 1 for AGILE full papers. The *Computational Environment* category shows the largest difference (although at overall low levels): AGILE scored better with a mean of 0.62 vs. 0.28 for GIScience. The *Results* category scores were again slightly higher for GIScience, with a mean of 1.05 vs. a mean of 0.88 for AGILE. Several papers in AGILE received a level of 0 here, indicating that crucial information is missing to connect analysis outputs and presented results. We refrain from comparing the *Preprocessing* category for the reasons stated earlier.

This comparison lets us draw two main conclusions: First, we conclude that both the target audience and the content of the two conference series are similar enough to be afflicted with similar shortcomings in terms of reproducibility, and thus, they both likely respond to similar solutions. Second, we conclude that the AGILE conference series seems structurally better positioned to support changing culture, because of a more stable audience and institutional support. The introduction of the AGILE reproducibility guidelines was achieved within a short time frame and with financial support in the form of an “AGILE initiative”, including travel funding for an in-person workshop. For GIScience, the task of changing the review process to foster better reproducibility falls squarely on the shoulders of the changing program committees. However, the initial results of AGILE’s new guidelines show that even small changes can lead to a significantly improved outcome.

## 6 Conclusions and outlook

In this work we investigated the reproducibility of several years of GIScience conference publications. The paper corpus is large enough for a representative sample and comparable to that used for the AGILE assessment study due to largely overlapping time window. However, this study does not intend to make judgements on AGILE vs. GIScience conference quality, nor to question the papers’ scientific soundness or relevance, since they were accepted for publication at a reputable conference. Instead, we investigated the papers along a single desirable quality dimension, reproducibility, which implies requirements on openness and transparency.

Using a similarly high bar for reproducibility as in the earlier assessment study, the results show room for improvement, as none of the presented articles were readily reproducible. The majority of articles provided some information, but not to the degree required to facilitate transparent and reusable research based on data and software. Overall, this is very similar to the outcomes of our earlier study on AGILE papers. As part of the AGILE assessment, we described concrete recommendations for individuals and organisations to improve paper reproducibility [26]. We have argued that AGILE and GIScience share a sufficiently common domain/discipline characteristics, audience, and author community, such that for both communities the strategies to improve the situation should be similar. Therefore, the previously identified recommendations are transferable to the GIScience conference series, with the most important recommendations being (1) promoting outstanding reproducible work, e.g., with awards or badges, (2) recognizing researchers’ efforts to achieve reproducibility, e.g., with a special track for reproducible papers, implementing a reproducibility review, open educational resources, and helpful author guidelines including data and software citation requirements and a specific data/software repository, and (3) making an institutional commitment to a policy shift that goes beyond mere accessibility [33]. These changes require a clear roadmap with a target year, e.g., 2024, when GIScience starts to only accept computationally reproducible submissions and to check reproducibility before papers are accepted. The concluding statement of Archmiller et al. [1] is directly transferable

to GIScience: The challenges are not insurmountable, and increased reproducibility will ensure scientific integrity. The AGILE reproducible paper guidelines [24] and the associated reproducibility review processes as well as other community code review systems such as CODECHECK [9] are open and “ready to use”. They can also be adopted for GIScience conferences, e.g., to suit the peer review process goals and scheduling. Kedron et al. [13] stressed the need for a comprehensive balanced approach to technical, conceptual, and practical issues. They further pointed out that simple availability does not automatically lead to adoption. Therefore, a broad discourse around these recommendations, tools, and concepts would be beneficial for all members of the community, whether their work is more towards conceptual, computational, or applied GIScience. A survey for authors, as conducted for AGILE [26], could help identify special requirements and specific circumstances, beyond the findings presented here and in related work.

Future work may replicate the reproducibility assessment at other major events and outlets for GIScience research, such as GeoComputation or COSIT conferences and domain journals (cf. [8] for an extensive list), but we would not expect significantly differing results. Practical reproductions of papers, and even more so replications of fundamental works, are promising projects to convincingly underpin a call for a culture change [29]. A successful *reproducibility turn* would not mean that every reproducible paper would be fully reproduced, nor would this be necessary. But at least for influential, e.g., highly cited papers, a validation of their applicability and transferability to other study areas should be possible – reproducibility is a prerequisite for that. For example, Egenhofer et al. [8] provide for a list of the most frequently cited articles as potential candidates. Such a project would ideally be supported with proper funding. There is currently growing activity in the GIScience discipline to address reproducibility and replicability of geospatial research. The GIScience conference community has the opportunity to play a leading and shaping role in this process, thereby ensuring its continuing attractiveness for authors to submit their work, and in consequence its high relevance for the wider GIScience discipline. A timely adoption of the technological and procedural solutions may allow GIScience researchers, together with the entirety of academia, to level up and approach the challenges of the “*second phase of reproducible research*” by tackling long-term funding for maintenance of code and data and building supporting infrastructure for reproducible research [31].

---

## References

- 1 Althea A. Archmiller, Andrew D. Johnson, Jane Nolan, Margaret Edwards, Lisa H. Elliott, Jake M. Ferguson, Fabiola Iannarilli, Juliana Vélez, Kelsey Vitense, Douglas H. Johnson, and John Fieberg. Computational Reproducibility in The Wildlife Society’s Flagship Journals. *The Journal of Wildlife Management*, 84(5):1012–1017, 2020. doi:10.1002/jwmg.21855.
- 2 Lorena A. Barba. Terminologies for Reproducible Research. *arXiv:1802.03311 [cs]*, 2018. arXiv: 1802.03311. URL: <https://arxiv.org/abs/1802.03311>.
- 3 Chris Brunsdon. Quantitative methods I: Reproducible research and quantitative geography. *Progress in Human Geography*, 40(5):687–696, 2016. doi:10.1177/0309132515599625.
- 4 Chris Brunsdon and Alexis Comber. Opening practice: supporting reproducibility and critical spatial data science. *Journal of Geographical Systems*, August 2020. doi:10.1007/s10109-020-00334-2.
- 5 Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker, and Barbara McGillivray. The citation advantage of linking publications to research data. *PLOS ONE*, 15(4):e0230416, 2020. doi:10.1371/journal.pone.0230416.
- 6 David L. Donoho. An invitation to reproducible computational research. *Biostatistics*, 11(3):385–388, July 2010. doi:10.1093/biostatistics/kxq028.

- 7 Matt Duckham, Edzer Pebesma, Kathleen Stewart, and Andrew U. Frank, editors. *Geographic Information Science*. Springer International Publishing, 2014. doi:10.1007/978-3-319-11593-1.
- 8 M. Egenhofer, K. Clarke, S. Gao, Teriitutea Quesnot, W. Franklin, M. Yuan, and David Coleman. Contributions of GIScience over the past twenty years. In Harlan Onsrud and Werner Kuhn, editors, *Advancing Geographic Information Science: The Past and Next Twenty Years*. GSDI Association Press, Needham, MA, 2016. URL: <http://www.gsdiassociation.org/images/publications/AdvancingGIScience.pdf>.
- 9 Stephen Eglén and Daniel Nüst. CODECHECK: An open-science initiative to facilitate sharing of computer programs and results presented in scientific publications. *Septentrio Conference Series*, (1), 2019. doi:10.7557/5.4910.
- 10 Juliana Freire, Norbert Fuhr, and Andreas Rauber. Reproducibility of Data-Oriented Experiments in e-Science (Dagstuhl Seminar 16041). *Dagstuhl Reports*, 6(1):108–159, 2016. doi:10.4230/DagRep.6.1.108.
- 11 Michael F. Goodchild. Geographical information science. *International journal of geographical information systems*, 6(1):31–45, 1992. doi:10.1080/02693799208901893.
- 12 Daniel S. Katz, Neil P. Chue Hong, Tim Clark, August Muench, Shelley Stall, Daina Bouquin, Matthew Cannon, Scott Edmunds, Telli Faez, Patricia Feeney, Martin Fenner, Michael Friedman, Gerry Grenier, Melissa Harrison, Joerg Heber, Adam Leary, Catriona MacCallum, Hollydawn Murray, Erika Pastrana, Katherine Perry, Douglas Schuster, Martina Stockhause, and Jake Yeston. Recognizing the value of software: a software citation guide. *F1000Research*, 9:1257, January 2021. doi:10.12688/f1000research.26932.2.
- 13 Peter Kedron, Wenwen Li, Stewart Fotheringham, and Michael Goodchild. Reproducibility and replicability: opportunities and challenges for geospatial research. *International Journal of Geographical Information Science*, 0(0):1–19, 2020. Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/13658816.2020.1802032>. doi:10.1080/13658816.2020.1802032.
- 14 Karen Kemp, Werner Kuhn, and Christoph Brox. Results of a survey to rate GIScience publication outlets. Technical report, AGILE Initiative - GIScience Publication Rating, 2013. URL: [https://agile-online.org/conference\\_paper/images/initiatives/results\\_of\\_a\\_survey\\_to\\_rate\\_giscience\\_publications.pdf](https://agile-online.org/conference_paper/images/initiatives/results_of_a_survey_to_rate_giscience_publications.pdf).
- 15 Carsten Keßler, Krzysztof Janowicz, and Tomi Kauppinen. spatial@linkedscience – Exploring the Research Field of GIScience with Linked Data. In Ningchuan Xiao, Mei-Po Kwan, Michael F. Goodchild, and Shashi Shekhar, editors, *Geographic Information Science*, Lecture Notes in Computer Science, pages 102–115, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-33024-7\_8.
- 16 Markus Konkol, Christian Kray, and Max Pfeiffer. Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study. *International Journal of Geographical Information Science*, 33(2):408–429, February 2019. doi:10.1080/13658816.2018.1508687.
- 17 Christian Kray, Edzer Pebesma, Markus Konkol, and Daniel Nüst. Reproducible Research in Geoinformatics: Concepts, Challenges and Benefits (Vision Paper). In Sabine Timpf, Christoph Schlieder, Markus Kattenbeck, Bernd Ludwig, and Kathleen Stewart, editors, *COSIT 2019*, volume 142 of *LIPICs*, pages 8:1–8:13. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.COSIT.2019.8.
- 18 Lawrence, Bryan, Jones, Catherine, Matthews, Brian, Pepler, Sam, and Callaghan, Sarah. Citation and Peer Review of Data: Moving Towards Formal Data Publication. *International Journal of Digital Curation*, 6(2), 2011.
- 19 Florian Markowetz. Five selfish reasons to work reproducibly. *Genome Biology*, 16:274, 2015. doi:10.1186/s13059-015-0850-7.
- 20 Jennifer A. Miller, David O'Sullivan, and Nancy Wiegand, editors. *Geographic Information Science*. Springer International Publishing, 2016. doi:10.1007/978-3-319-45738-3.

- 21 Jannes Muenchow, Susann Schäfer, and Eric Krüger. Reviewing qualitative GIS research—Toward a wider usage of open-source GIS and reproducible research practices. *Geography Compass*, 13(6):e12441, 2019. doi:10.1111/gec3.12441.
- 22 Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware, and John P. A. Ioannidis. A manifesto for reproducible science. *Nature Human Behaviour*, 1:0021, January 2017. doi:10.1038/s41562-016-0021.
- 23 Brian A. Nosek and Timothy M. Errington. What is replication? *PLOS Biology*, 18(3):e3000691, March 2020. doi:10.1371/journal.pbio.3000691.
- 24 Daniel Nüst, Frank Ostermann, Rusne Sileryte, Barbara Hofer, Carlos Granell, Marta Teperek, Anita Graser, Karl Broman, and Kristina Hettne. AGILE Reproducible Paper Guidelines, 2019. doi:10.17605/OSF.IO/CB7Z8.
- 25 Daniel Nüst, Frank Ostermann, Rusne Sileryte, Barbara Hofer, Carlos Granell, Marta Teperek, Anita Graser, Karl Broman, and Kristina Hettne. Reproducible Publications at AGILE Conferences, 2019. doi:10.17605/OSF.IO/PHMCE.
- 26 Daniel Nüst, Carlos Granell, Barbara Hofer, Markus Konkol, Frank O. Ostermann, Rusne Sileryte, and Valentina Cerutti. Reproducible research and GIScience: an evaluation using AGILE conference papers. *PeerJ*, 6:e5072, 2018. doi:10.7717/peerj.5072.
- 27 Daniel Nüst, Frank Ostermann, Carlos Granell, and Barbara Hofer. Reproducibility package for "Reproducible Research and GIScience: an evaluation using GIScience conference papers", September 2020. doi:10.5281/zenodo.4032875.
- 28 Daniel Nüst, Frank Ostermann, Carlos Granell, and Alexander Kmoch. Improving reproducibility of geospatial conference papers – lessons learned from a first implementation of reproducibility reviews. *Septentrio Conference Series*, (4), September 2020. doi:10.7557/5.5601.
- 29 Frank O. Ostermann. Linking Geosocial Sensing with the Socio-Demographic Fabric of Smart Cities. *ISPRS International Journal of Geo-Information*, 10(2):52, January 2021. doi:10.3390/ijgi10020052.
- 30 Roger D. Peng. Reproducible Research in Computational Science. *Science*, 334(6060):1226–1227, December 2011. doi:10.1126/science.1213847.
- 31 Roger D. Peng and Stephanie C. Hicks. Reproducible Research: A Retrospective. *arXiv:2007.12210 [stat]*, July 2020. arXiv: 2007.12210. URL: <http://arxiv.org/abs/2007.12210>.
- 32 James H. Stagge, David E. Rosenberg, Adel M. Abdallah, Hadia Akbar, Nour A. Attallah, and Ryan James. Assessing data availability and research reproducibility in hydrology and water resources. *Scientific Data*, 6(1):190030, February 2019. Number: 1 Publisher: Nature Publishing Group. doi:10.1038/sdata.2019.30.
- 33 Victoria Stodden, Jennifer Seiler, and Zhaokun Ma. An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589, 2018. doi:10.1073/pnas.1708290115.
- 34 S. Winter, A. Griffin, and M. Sester, editors. *Proceedings 10th International Conference on Geographic Information Science (GIScience 2018)*, volume 114. LIPIcs, 2018. URL: <http://www.dagstuhl.de/dagpub/978-3-95977-083-5>.
- 35 Ningchuan Xiao, Mei-Po Kwan, Michael F. Goodchild, and Shashi Shekhar, editors. *Geographic Information Science*. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-33024-7.

# Comparison of Simulated Fast and Green Routes for Cyclists and Pedestrians

Christina Ludwig ✉ 

GIScience Research Group, Institute of Geography, Heidelberg University, Germany

Sven Lautenbach ✉ 

Heidelberg Institute for Geoinformation Technology (HeiGIT) gGmbH at Heidelberg University, Germany

Eva-Marie Schömann ✉

GIScience Research Group, Institute of Geography, Heidelberg University, Germany

Alexander Zipf ✉ 

GIScience Research Group, Institute of Geography, Heidelberg University, Germany  
HeiGIT gGmbH at Heidelberg University, Germany

---

## Abstract

Routes with a high share of greenery are attractive for cyclist and pedestrians. We analyze how strongly such green routes differ from the respective fast routes using the openrouteservice. Greenness of streets was estimated based on OpenStreetMap data in combination with Sentinel-II imagery, 3d laser scan data and administrative information on trees on public ground. We assess the effect both at the level of the individual route and at the urban level for two German cities: Dresden and Heidelberg. For individual routes, we study how strongly green routes differ from the respective fast routes. In addition, we identify parts of the road network which represent important green corridors as well as unattractive parts which can or cannot be avoided at the cost of reasonable detours. In both cities, our results show the importance of urban green spaces for the provision of attractive green routes and provide new insights for urban planning by identifying unvegetated bottlenecks in the street network for which no green alternatives exist at this point.

**2012 ACM Subject Classification** Computing methodologies → Simulation evaluation; General and reference → Empirical studies; Applied computing → Cartography

**Keywords and phrases** Routing, OpenStreetMap, route choice, urban vegetation, sustainable mobility

**Digital Object Identifier** 10.4230/LIPICs.GIScience.2021.II.3

**Supplementary Material** The source code to apply this analysis to other geographic regions can be found at <https://doi.org/10.5281/zenodo.5483586>.

**Funding** This research was funded by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) in the context of the research initiative mFUND (grant number 19F2073A).  
*Sven Lautenbach:* Acknowledges funding by the Klaus-Tschira Stiftung.

**Acknowledgements** We would like to thank several research assistants involved in the project, namely Anna Lausen, Pascal Wolf as well as Robert Klar and Max Girmann who wrote their theses in the context of healthy routing. We acknowledge support by the HeiGIT openrouteservice team, especially by Adam Rousell and Andrzej Oleś as well as by the HeiGIT big spatial data team. Furthermore, we would like to thank the meinGrün project team (<http://meingruen.ioer.info/>) for fruitful discussions and the city administration of Dresden and Heidelberg for their support.

## 1 Introduction

Route preferences of pedestrians and cyclists differ from the ones of car drivers. In addition to distance and time effort, aspects such as slope, safety, aesthetics, presence of green areas, accessibility to locations of interest, avoidance of noise and air pollution, road surface and heat stress play a role [1, 30, 33, 34, 35, 38]. Several of these factors have been integrated



© Christina Ludwig, Sven Lautenbach, Eva-Marie Schömann, and Alexander Zipf;  
licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 3; pp. 3:1–3:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

as routing weights in specialized navigation systems [41, 26, 34, 23]. We focus here on green routes – i.e. routes with a high share of vegetation alongside of them. Urban green spaces have been shown to clearly benefit mental and physical well-being of city dwellers [37, 15, 12, 20, 18] in addition to providing other important ecosystem services [8] and space for recreational and cultural activities such as sports, experiencing nature or social exchange [5, 16, 31]. These effects are not limited to public green spaces such as parks but also include urban greenery along roads and paths.

Including additional factors such as greenness in the route generation process will in most situations increase the travel distance and time, which in many situations is still the most important factor in route choice. Therefore, it is necessary for the design of navigation apps and planning support systems to study these trade-offs and to adjust weighting factors so that reasonable and attractive route recommendations are generated.

Previous studies have analyzed different routing options for pedestrians and cyclists based on different factors such as cycling infrastructure, traffic volume or scenery [13, 21]. Novack et al [23] has analyzed one thousand randomly chosen routes with respect to detour factors with respect to green, noise-avoiding and social routes for pedestrians at the example of the city of Heidelberg. We extend this work here with respect to the following aspects: i) we analyze routes for two cities – Heidelberg and Dresden –, ii) we generate a larger set of routes that connect different quarters of the cities and iii) we analyze detour factors both for pedestrians as well as for cyclists. In addition to the evaluation from the perspective of the individual user (“what is the best route from A to B?”) we also analyze the routes from the perspective of city planners to identify parts of the traffic network where attractive, green routes exist and where they are missing.

The aims of the analysis at the example of the two cities are twofold:

1. How do the green routes differ from the respective fast routes? (RQ1)
2. Which parts of the citywide traffic network represent important, green corridors, which parts can be potentially avoided due to low greenness and which parts cannot be avoided since no green alternatives exist? (RQ2)

We aim at a systematic analysis of the effects of greenness on the individual routes and a quantification of the availability of green routes for cyclists and pedestrians in both cities.

## **2** Methods and data

### **2.1** Study sites

The analysis was performed in two German cities: Heidelberg and Dresden. Dresden is the capital of the federal state of Saxony and is located in the eastern part of Germany with a population of 563,011 (2019) and a total area of 328.8  $km^2$ . Heidelberg is smaller in size (108,8  $km^2$ ) and population (160,355 in 2019). Both cities are traversed by large rivers - the Elbe in Dresden and the Neckar in Heidelberg - which puts constraints on the connectivity between several city quarters. Restrictions also arise due to the terrain, especially in the case of Heidelberg where the low mountain range Odenwald reaches into the city limits. Both cities provide extended green spaces such as municipal parks, floodplains along the rivers, large forest areas at the city borders and private gardens within residential areas. An estimated 74% of Heidelberg and 66% of Dresden are covered by vegetation based on land cover map by DLR (2020) [7]. As most German cities, cycling infrastructure is present and cycling is encouraged by the city administration.

## 2.2 Methods

### 2.2.1 OSM Data Quality Assessment

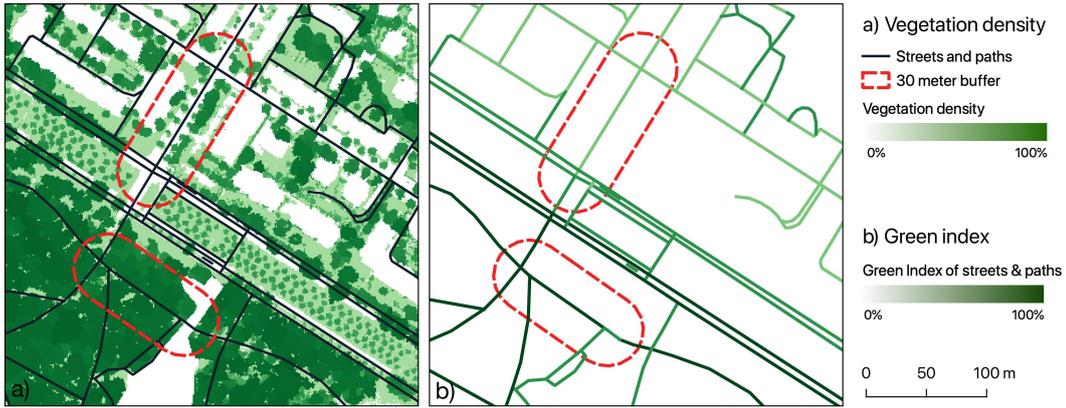
OpenStreetMap (OSM) was used as the data source for the road network, the cycling infrastructure and partly for the identification of urban vegetation. Its benefits are its open licence and its increasing availability all over the world. Since OSM data is produced by local and remote volunteers, the data quality varies spatially and depending on the thematic domain. Therefore, it is important to check data quality before using it in scientific analyses and other applications. We followed an intrinsic data quality approach [32, 4, 2] and queried contributions for different *highway* categories and *cycleways* and studied the development of contributions over time. OSM history was accessed via the ohsome API that builds on the OpenStreetMap History Database (OSHDB) [28]. We analyzed the temporal evolution of the overall length of all OSM objects belonging to relevant *highway* classes. In addition, we analyzed the number of OSM users in Heidelberg and Dresden that contributed actively in the domain of roads and paths (*highway=\**). The analysis of the data quality was performed in R [27] using the packages sf [25], RCurl [36], geojsonio [3], tidyverse [40], ggplot2 [39] and ggpubr [14].

### 2.2.2 General Routing

The pedestrian and bike routes were calculated using the openrouteservice (ORS), an open source routing engine which is based on OSM data [22]. Its source code is publicly available on Github (<https://github.com/GIScience/openrouteservice>) and it can be used for worldwide routing queries through its public API or web client (<https://maps.openrouteservice.org>). The routing function of the ORS provides different user profiles such as car, bike or pedestrian as well as other query parameters (e.g. avoid ferries) to adapt the route to different user preferences. By integrating additional data in the route calculation, the ORS can generate specialized routes such as green routes, which prefer paths along greenery, or quiet routes, which prefer paths with less traffic noise. For this study, a new instance of the ORS was created which uses region specific data for the greenness of streets and uses a different weighting function to balance greenness, steepness and duration of the route.

In general, routing algorithms are based on an undirected graph consisting of edges representing streets or paths and nodes representing intersections. Every edge in the graph is assigned a weight, e.g. the street length to find the shortest route or the travel time to find the fastest route. Since the travel speeds and preferences of pedestrians and cyclists are different, separate graphs are built for each of them. The optimal route between two points, e.g. the fastest or shortest, is the route with the lowest overall weight. There are different algorithms to find this optimal route such as the Dijkstra's algorithm [6], which is also offered by the ORS but other, more efficient algorithms are preferred. In this study, the route calculation was based on the A\* algorithm [10].

The default route recommended by the ORS is the fastest route with some restrictions based on the street types to make the route more attractive, e.g. avoiding unpaved paths for cyclists or considering the fact that cyclists may need to dismount their bikes and walk on designated foot paths. In the remainder of the text we refer to this route as the fast route. The same restrictions on street or path types were considered in all our simulations - i.e. also for the green routes.



■ **Figure 1** Calculation of the green index at the example of a subset in Dresden where the estimation was based on vegetation density data derived from 3D laser scan data. The green index is the share of vegetation in a 30m buffer around each street segment.

### 2.2.3 Green Routing

Conventional routing algorithms only use the distance or travel time of the route segments as weights in the routing graph to find the optimal route. In order to calculate a green route, these weights have to be adapted by integrating the greenness of each street segment into their calculation. This information is not given in OSM directly but needs to be derived prior to building the routing graph. Since data availability differed in Dresden and Heidelberg, two different approaches were used. In Dresden, data on vegetation density with a spatial resolution of 1 meter derived from 3D laser scan data was used [11]. In Heidelberg, the vegetation presence was quantified by combining OSM data, Sentinel-2 imagery and municipal tree cadastre data. Information from OSM and Sentinel-2 data were fused using the Dempster-Shafer Theory as described in Ludwig et al. [19] to yield a map of vegetation presence which was later enriched by single trees extracted from OSM and in the municipal tree cadastre data. In the end, the greenness of a certain street segment - called green index from now on - was quantified by calculating the fraction of area covered by vegetation with a 30 meter buffer around the street segment (c.f. Figure 1). Performing a visibility analysis to only consider vegetation which is visible from the street and not obscured by buildings was tested but had to be excluded eventually due to the high computational costs.

For the green routing, the weights of the edges in the routing graph were calculated using

$$w_i = d_i + \frac{((1 - g_i) * w_g + s_i)}{2} * d_i \quad (1)$$

where  $w_i$  is the weight of street segment  $i$ ,  $d_i$  is the time needed to traverse it,  $g_i$  is its green index,  $s_i$  is its steepness index and  $w_g$  is the green weight factor with range  $[0,1]$ , which controls the trade-off between greenness and duration of the route. The higher this green weight factor, the greener but also the longer the route. The green index was converted to a cost factor using  $1 - g_i$ , so that streets with low greenness have higher weights and are thus avoided. Since the green index and steepness index represent intensities, it was multiplied by the travel duration of the respective route segment to yield an exposure value.

The steepness was included as an additional cost factor in the function to avoid yielding routes with high ascents and descents. This was done, because currently the ORS does not consider the slope in the calculation of the travel time, e.g., by assuming slower travel speeds for ascending route segments. The influence of the steepness is defined by the *steepness difficulty* parameter of the ORS which was set to 1 (*moderate*) in this study.

In this study, different green routes were compared to their respective fast routes. The fast routes did not consider greenness, i.e. their green weight factor was  $w_g = 0$ . Green routes were created using green weight factors ranging between 0.1 and 1 ( $0.1 \leq w_g \leq 1$ ). The higher this value, the greener and longer the route. An example of a fast ( $w_g = 0$ ) and a green route ( $w_g = 1$ ) for pedestrians is given in figure 2. While the fast route (22:50 minutes, 1.9 km) in the example follows the main road through a built up area, the green route (26:20 minutes, 2.2 km) leads through natural and agricultural fields. The green route was 15% longer (3:30 minutes) but its greenness value was almost three times higher (260%) compared to the fast route.



■ **Figure 2** Example of a fast and green pedestrian route in Heidelberg. The green route (26:20 minutes, 2.2 km) is 15% longer (3:30 min) than the fast route (22:50 minutes, 1.9 km) but its green value is almost three times higher (260%) compared to the fast route.

## 2.2.4 Route simulations

To answer the research questions, random trips for pedestrians and cyclists were simulated for Heidelberg and Dresden by generating random start and end points within the respective administrative boundaries. To simulate the trips according to the distribution of the population density the trips were generated based on the city districts with each trip starting in one district and ending in another one. In addition, short trips were simulated by generating random start and end points within the same district. In this way, more routes were generated in the populated city center and less routes in the unpopulated outskirts (e.g. forests). For pedestrian routes, the minimum length was set to 200 meters and the maximum length to 5 kilometers to represent both short trips of everyday life (e.g. grocery shopping) as well as longer recreational trips (e.g. taking a walk, touristic walking tours). The minimum and maximum lengths of cycling trips were set to 500 meters and 15 kilometers respectively.

In very rare cases the computed duration of the green route was lower than the fast route, since the terrain was only considered through the steepness index and not in the calculation of the route duration. This was for example the case when a green route lead through the forested low mountain range close to Heidelberg, while the fast route went around it yielding a longer route. This phenomenon was due to the fact that the very high greenness in the forest made up for the high steepness in the routing costs. Since such green alternative routes might not be seen as attractive alternatives in reality, they were dismissed and another random trip was generated.

Two separate simulation runs were performed to answer the two research questions. The first run was performed to compare the green routes to their respective fast routes in terms of duration and similarity (RQ1). For each city, 3000 trips by bike and 3000 trips by foot were simulated. For each trip, the fast route and 10 green routes with increasing green weight factor  $w_g$  in the range of 0.1 to 1.0 were calculated.

A second simulation run was performed to analyze the street network connectivity in regard to the exposure of cyclists and pedestrians to greenness (RQ2). This analysis was performed in Dresden by simulating 8858 bike trips and in Heidelberg by simulating 9024 trips for pedestrians. For each trip, the fast route and an alternative green route with green weight factor  $w_g = 1$  were calculated and compared to each other. Avoidable street segments were identified by extracting the route segments of the fast route which were not identical with the green route. Preferable street segments were identified by extracting the route segments of the green route which were not identical with the fast route. Unavoidable route segments were defined as streets or paths which were contained in both the fast and green route and showed a green index lower than 0.1. Streets where the fast and green route were identical and which had a green index higher than 0.6 were considered as ideal route segments, i.e. the route is both fast and green. The extracted route segments were aggregated into heatmaps showing avoidable, preferable, unavoidable and ideal street or path segments. The heatmaps were created using the Kernel Density Estimation module of SAGA GIS version 2.3.2 (<http://www.saga-gis.org/en/index.html>). Since this tool requires point data as input, the extracted route segments were converted from line to point geometries with a distance of 5 meters between the points. The rest of the analysis was implemented using Python 3.9 (<https://www.python.org/>) and the openrouteservice Python package (<https://pypi.org/project/openrouteservice/>).

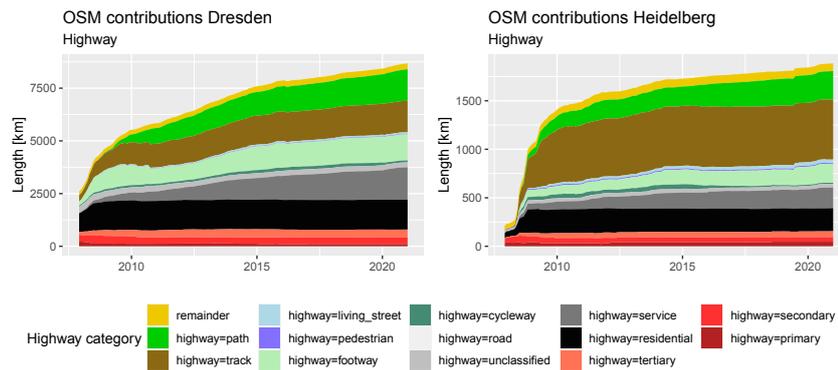
## 3 Results

### 3.1 OSM Data Quality Assessment

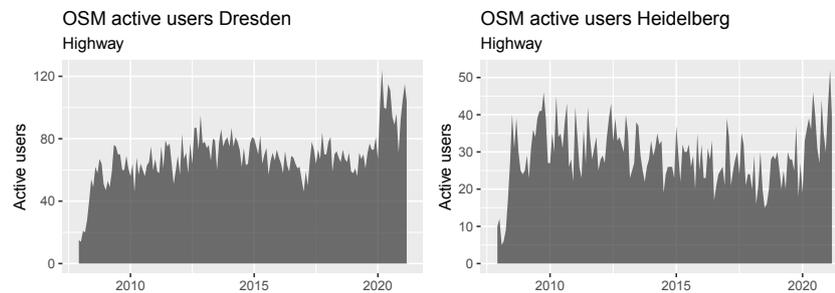
Most road classes in both Heidelberg and Dresden showed a contribution pattern that indicates saturation (c.f. figure 3) – together with the constantly high user activity (c.f. figure 4) in both cities this can be taken as an indicator for high completeness of the road network. Road categories that could be assumed to be highly complete in both cities consist of: *primary*, *secondary*, *tertiary*, *residential*, *unclassified* and *pedestrian* roads. In addition, the category *living\_street* seemed saturated in Dresden, while in Heidelberg this was true for the category *tracks*.

Two road categories were declining in both cities: *roads* and *cycleways*. *Roads* are a category used if a road cannot be categorized clearly into one of the other tags so a decline of this category is both expected and an indication of increasing data quality. The category *cycleway* represents separate ways for cyclists. Cycleways that are not separated from a road are usually mapped with a *highway=\** tag and the additional key *cycleway=\** that specifies the type of the cycleway. The use of the tag *cycleway=\** has evolved over time leading to a decrease in the use of *highway=cycleway*. The length of ways tagged as *cycleway* has increased in both cities over time which can be taken as an indication that cycleways are relatively complete (c.f. 5).

The road categories *path*, *service* and *footway* seemed not yet saturated. In Heidelberg in addition the category *living\_street* and in Dresden *track* were not saturated. The different highway values are explained in the OSM Wiki (<https://wiki.openstreetmap.org/wiki/Key:highway>).



■ **Figure 3** Development of highway contributions in OSM for the two case studies. The category remainder captures all remaining road classes.



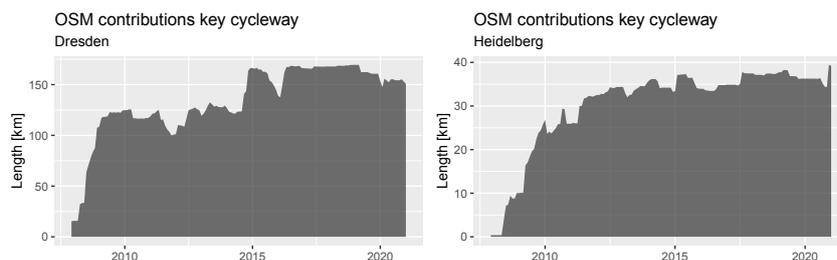
■ **Figure 4** Active OSM users that contributed highway features for the two case studies.

### 3.2 Route Duration vs. Route Greenness

As expected, the route simulations showed that the duration of the green route increased with an increasing green weight factor. Compared to the fast route, 95% of green cycling routes with  $w_g = 1$  were up to 21% longer in both cities. For pedestrian routes, this value was at 16% in Dresden and 13% in Heidelberg. Since detours of this magnitude have also been reported in other studies which have analyzed the route choice behavior (e.g. [24]) these simulated greener alternative routes can be seen as reasonable. Green routes which showed a longer detour than this usually led through forested or agricultural areas bordering the outskirts of the cities, so there was a considerable gain in greenness justifying the bigger detour.

The increase in duration also depended on the overall length of the trip (Figure 6). For bike trips in Dresden which took up to 10 minutes, the green route was on average only 8 seconds (1.7%) longer than the fast one, while for trips which took 20-30 minutes the increase in travel time was about 1:32 minutes (6.1%). In both cities, the increase in duration of the green route was lower for pedestrian trips than for cycling trips, which might be due to the fact that the pedestrian routes are shorter leaving less options for detours.

For both pedestrian and cycling routes, the gain in greenness exceeded the increase in duration on a percentage basis. The gain in greenness relative to the increase in duration was at 3.39 for cycling routes in Dresden, i.e. for every 1% increase in duration the routes were 3.39% greener (Figure 7a). In Heidelberg, this value was even at 5.26% (Figure 7c), which is probably due to the fact that many green routes connecting the city center with



■ **Figure 5** Development of cycleway contributions in OSM for the two case studies. The figure does not reflect that cycleways can be on one or both sides of the street but tagged on a single way.

districts in the eastern part of the city lead through the Odenwald forest. These alternative routes are often a lot greener but also steeper, which is why they cannot always be seen as valid alternatives in reality - at least if the purpose of the trip is not sporting activity. For pedestrians, the relative gain in greenness compared to trip duration was also lightly higher in Heidelberg than in Dresden (Figure 7b+d).

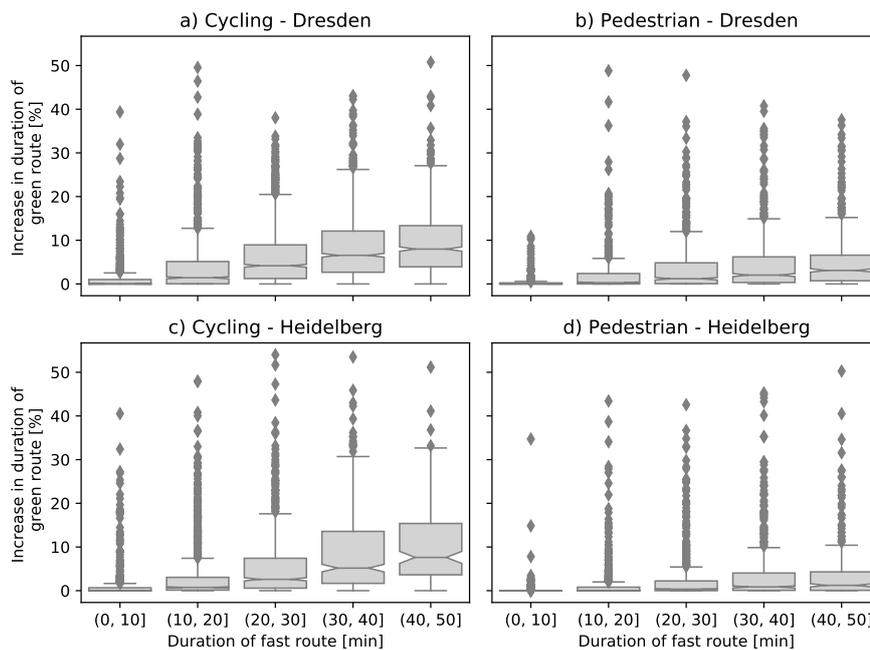
With an increasing green factor, the green route also deviated stronger from the fast route. These deviations were generally higher for cycling routes than for pedestrian routes (Figure 8). This might be due to the fact that cycling routes were on average longer than pedestrian routes giving them more options for alternative routes. The deviations were strongest for cyclists in Dresden where green routes with the highest green factor  $w_g = 1$  were on average 51% identical with the fast route (Figure 8a). In Heidelberg, the fast and green cycling routes were more similar sharing on average 59% of the route (Figure 8c).

### 3.3 Analysis of the road network with regard to availability of green routing alternatives

In Dresden the fast cycling routes mostly lead along the main streets, while the green alternative routes lead along the Elbe river or through public green spaces such as the Great Garden (in the south east of Figure 9). The later one seems to be a very valuable connection in the bike network, since it is both the fastest and greenest route to choose for many bike trips. Bridges across railways on the other hand often show low vegetation and cannot be avoided easily by making reasonable detours (marked in purple in Figure 9). Upgrading these street segments could increase the quality of the whole cycling network.

A comparison between the avoidable and preferable greener routes in Dresden and real cycling traffic data collected from 2018 until 2020 [9] indicated that the preferable greener cycling routes were also frequently used by cyclists in reality (Figure 10). For example, the cycling path along the Elbe river and through the Great Garden - which were identified as preferred segments in our analysis - showed high numbers of cyclists in reality. Still, some routes, which according to the simulated data could be avoided in favour of greener routes, were still quite highly frequented in reality, e.g. the main road leading from East to West in the lower part of Figure 10.

In Heidelberg, the avoidable and preferable paths were analyzed for pedestrians (Figure 11). The main roads leading through built-up areas were avoided while routes through green spaces and along agricultural fields were preferred by the green routes compared to the fast routes. The Ochsenkopfwiese for example, a public green space in the center of Heidelberg (Figure 11 center), was supposed to be replaced by a bus depot until the plans were stopped



■ **Figure 6** Increase in duration of green routes compared to fast routes depending on trip duration. Trips were grouped based on duration of the fast route in bins of 10 minute intervals.

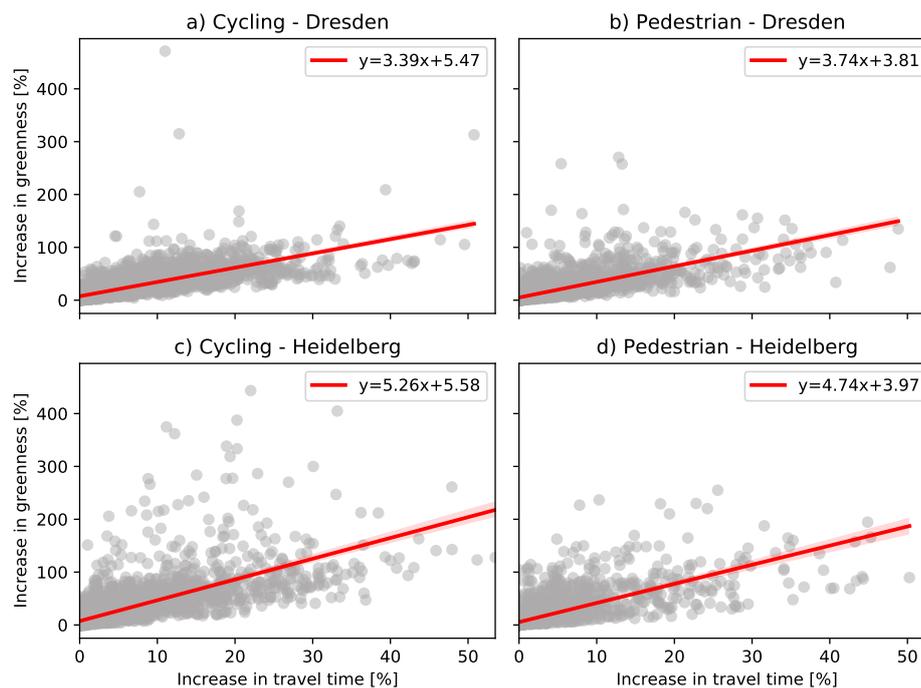
by a referendum. The results of this analysis confirm how important the preservation of this green space is, since it is one of several important green corridors that enable citizens to move through the city on attractive routes. Similar to Dresden, bridges and overpasses, such as the one close to the main train station cannot be avoided easily due to the lack of green alternative routes.

## 4 Discussion

The simulations have shown that in many cases it should be possible for both pedestrians and cyclists to take greener, more attractive routes by taking reasonable detours. Especially urban green spaces and paths along rivers represent important green corridors in cities which should be preserved. Bridges without any greenery on the other hand often form bottlenecks where no reasonable green alternatives exist. Enhancing greenness in these places by the city administration could increase satisfaction of cyclists and pedestrians and stimulate the use of sustainable traffic modes.

The validity of these results was supported by the comparison to real cycling traffic data in Dresden, which showed that most of the preferable green routes were also highly frequented in reality. However, the fact that some avoidable routes identified in our simulation were still frequently used in reality suggests that there are other factors in the route choice of cyclists which have not been considered in this study such as perceived safety or quality of the bike infrastructure. These aspects should be investigated in future studies. In addition, other quality metrics such as the continuity of the route as described by [29] could be included in the comparison of the routes.

There are some improvements to the routing algorithm which should be considered in future studies. First, the terrain should be considered in the calculation of the route duration to make sure to generate attractive alternative routes without high ascents. Second, other

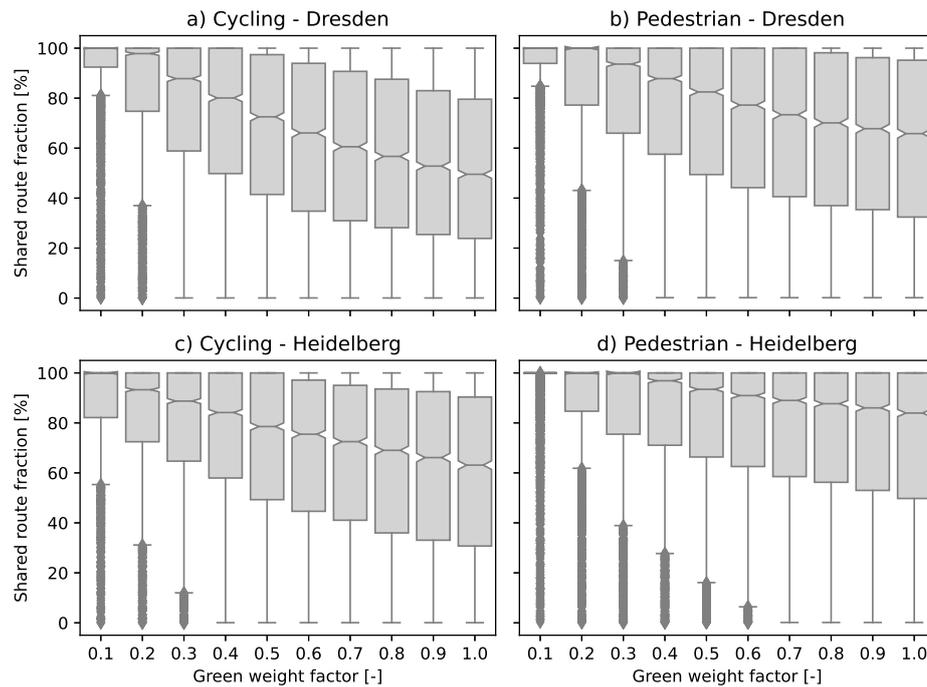


■ **Figure 7** Increase in greenness compared to the increase in travel time of simulated green cycling (a+c) and pedestrian (b+d) routes in Heidelberg and Dresden. The red line is based on a linear model, shown together with the 95%-confidence band.

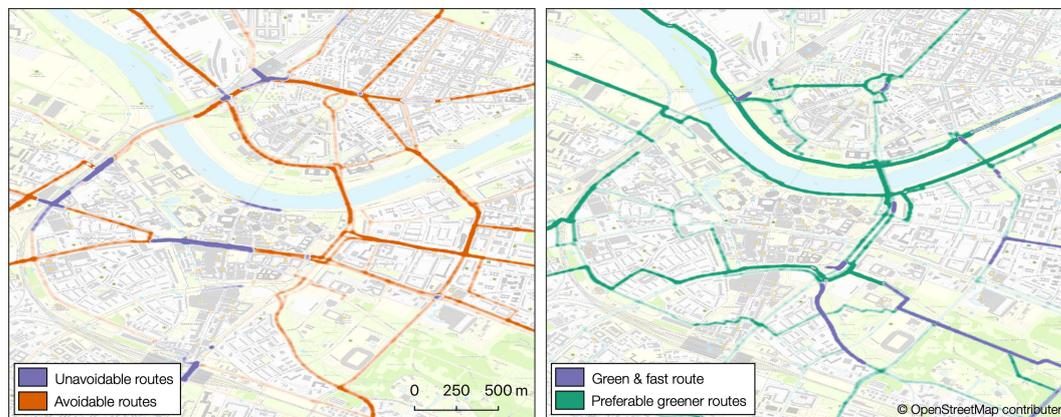
routing algorithms such as evolutionary or k-shortest path algorithms should be evaluated for their potential to generate multiple pareto-optimal route options which might not be found using the A\* or Dijkstra's algorithm.

Our two case studies represent two cycling friendly cities with relative large shares of private and public green. For cities with a different cycling infrastructure and green space distribution the trade-off between greenness and time effort might be stronger than in our case studies. Other factors of relevance are presumably the terrain, presence of large water bodies and the general layout of the cities which further influence which pathways are of special importance in regard to the connectivity of the network. Therefore, the results of this analysis might not be directly transferable to any other city.

Still, given a sufficient completeness of roads, footpaths and cycling infrastructure in OSM a transfer of our green routing approach to other cities seems feasible. As done in this study, an OSM data quality analysis should be conducted prior to the analysis. In addition, the transferability of the method depends on the available data used to estimate the greenness of the streets. High quality vegetation density data derived from 3D laser scan data as used in Dresden are usually not available for most cities. However, a comparison of green routes in Dresden based on i) OSM in combination with Sentinel-II data and ii) the use of vegetation density data from 3D laser scans showed very similar results indicating that free and open data are sufficient to generate reasonable green routes if high quality data is missing [17]. In this way, the proposed approach could also be used to assist urban planners in identifying potential for improvement in the road network for pedestrians and cyclists even if no local data on their mobility patterns are available.



■ **Figure 8** Shared fraction between fast and green route dependent on the green weight factor  $w_g$ . With increasing green weight factor, the green and fast routes diverge from one another.



■ **Figure 9** Avoidable and unavoidable route segments for cyclists in Dresden due to lower greenness (left). Preferable greener alternative routes and ideal route segments (right).

## 5 Conclusion

For our two case studies, reasonable green routing alternatives exist for cyclists and pedestrians for most parts of the cities highlighting the importance of urban green spaces. We were further able to identify bottlenecks in the road network which should be prioritized by urban planning to further foster sustainable and healthy traffic. A comparison with observed cycling data suggest both that the approach is generally suitable but that additional factors should be taken into account.



■ **Figure 10** Comparison between our analysis (left) and observed cyclists (right) for Dresden. Greener alternative routes identified in the simulation are also highly frequented in reality.



■ **Figure 11** Unavoidable, avoidable and preferable greener alternative routes for pedestrians in Heidelberg. Green routes avoid the main roads leading through built up areas while preferring fields and urban green spaces.

Given the importance of pedestrian and cyclist traffic with respect to climate change adaptation and public health further effort will be undertaken to scale the presented green routing with the public ORS instance to much larger regions. Upcoming work will also focus the extension of the green routing approach by incorporating further health related factors such as the avoidance of heat stress.

---

## References

- 1 Sabreena Anowar, Naveen Eluru, and Marianne Hatzopoulou. Quantifying the value of a clean ride: How far would you bicycle to avoid exposure to traffic-related air pollution? *Transportation Research Part A: Policy and Practice*, 105:66–78, 2017. doi:10.1016/j.tra.2017.08.017.
- 2 Christopher Barron, Pascal Neis, and Alexander Zipf. A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, 18(6):877–895, 2014. doi: 10.1111/tgis.12073.
- 3 Scott Chamberlain and Andy Teucher. *geojsonio: Convert Data from and to 'GeoJSON' or 'TopoJSON'*, 2021. R package version 0.9.4. URL: <https://CRAN.R-project.org/package=geojsonio>.

- 4 Livia Castro Degrossi, João Porto de Albuquerque, Roberto dos Santos Rocha, and Alexander Zipf. A Framework of Quality Assessment Methods for Crowdsourced Geographic Information: a Systematic Literature Review. In *Proceedings of the 14th ISCRAM Conference*, pages 21–24, 2017. Issue: May.
- 5 Dawn C Dickinson and Richard J Hobbs. Cultural ecosystem services: Characteristics, challenges and lessons for urban green space research. *Ecosystem Services*, 25:179–194, 2017.
- 6 E W Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- 7 German Aerospace Center (DLR). Land Cover DE - Sentinel-2 - Germany, 2015, 2020. doi:10.15489/1ccmlap3mn39.
- 8 Erik Gómez-Baggethun, Åsa Gren, David N Barton, Johannes Langemeyer, Timon McPhearson, Patrick O’farrell, Erik Andersson, Zoé Hamstead, and Peleg Kremer. Urban ecosystem services. In *Urbanization, biodiversity and ecosystem services: Challenges and opportunities*, pages 175–251. Springer, Dordrecht, 2013. doi:10.1007/978-94-007-7088-1.
- 9 P. Grubitzsch, S. Lißner, S. Huber, and T. Springer, 2021. Accessed: 23.4.2021. URL: <https://www.mcloud.de/web/guest/suche/-/results/suche/relevance/MOVEBIS/0/detail/ECF9DF02-37DC-4268-B017-A7C2CF302006>.
- 10 Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. URL: <https://www.cs.auckland.ac.nz/compsci767s2c/projectReportExamples.d/astarNilsson.pdf>.
- 11 Robert Hecht, Gotthard Meinel, and Manfred F Buchroithner. Estimation of urban green volume based on single-pulse lidar data. *IEEE Transactions on Geoscience and Remote Sensing*, 46(11):3832–3840, 2008. doi:10.1109/TGRS.2008.2001771.
- 12 Victoria Houlden, Scott Weich, João Porto de Albuquerque, Stephen Jarvis, and Karen Rees. The relationship between greenspace and the mental wellbeing of adults: A systematic review. *PLOS ONE*, 13(9):e0203000, September 2018. doi:10.1371/journal.pone.0203000.
- 13 Isaac Johnson, Jessica Henderson, Caitlin Perry, Johannes Schöning, and Brent Hecht. Beautiful... but at what cost? an examination of externalities in geographic vehicle routing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–21, 2017. doi:10.1145/3090080.
- 14 Alboukadel Kassambara. *ggpubr: 'ggplot2' Based Publication Ready Plots*, 2020. R package version 0.4.0. URL: <https://CRAN.R-project.org/package=ggpubr>.
- 15 Michelle Kondo, Jaime Fluehr, Thomas McKeon, and Charles Branas. Urban green space and its impact on human health. *International Journal of Environmental Research and Public Health*, 15(3):445, 2018. doi:10.3390/ijerph15030445.
- 16 Kerstin Krellenberg, Juliane Welz, and Sonia Reyes-Päcke. Urban green areas and their potential for social interaction—a case study of a socio-economically mixed neighbourhood in Santiago de Chile. *Habitat International*, 44:11–21, 2014.
- 17 Anna Lausen. Analyse der Verwendbarkeit von OSM Daten für ein grünes Routing am Beispiel Dresden. Bachelor’s thesis, Heidelberg University, Institute of Geography, Heidelberg, Germany, January 2020.
- 18 Claudia Lega, Christopher Gidlow, Marc Jones, Naomi Ellis, and Gemma Hurst. The relationship between surrounding greenness, stress and memory. *Urban Forestry & Urban Greening*, 59:126974, 2021. doi:10.1016/j.ufug.2020.126974.
- 19 Christina Ludwig, Robert Hecht, Sven Lautenbach, Martin Schorcht, and Alexander Zipf. Mapping Public Urban Green Spaces Based on OpenStreetMap and Sentinel-2 Imagery Using Belief Functions. *ISPRS International Journal of Geo-Information*, 10(4):251, April 2021. doi:10.3390/ijgi10040251.
- 20 J. Maas. Green space, urbanity, and health: how strong is the relation? *Journal of Epidemiology & Community Health*, 60(7):587–592, July 2006. doi:10.1136/jech.2005.043125.

- 21 David Philip McArthur and Jinhyun Hong. Visualising where commuting cyclists travel using crowdsourced data. *Journal of transport geography*, 74:233–241, 2019. doi:10.1016/j.jtrangeo.2018.11.018.
- 22 Pascal Neis and Alexander Zipf. Openrouteservice.org is three times “open”: Combining OpenSource, OpenLS and OpenStreetMaps. In *GIS Research UK (GISRUK 08)*. Manchester, 2008.
- 23 Tessio Novack, Zhiyong Wang, and Alexander Zipf. A system for generating customized pleasant pedestrian routes based on openstreetmap data. *Sensors*, 18(11):3794, 2018.
- 24 Yujin Park and Gulsah Akar. Why do bicyclists take detours? a multilevel regression model using smartphone gps data. *Journal of transport geography*, 74:191–200, 2019.
- 25 Edzer Pebesma. Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1):439–446, 2018. doi:10.32614/RJ-2018-009.
- 26 Andrea Pimpinella, Alessandro E.C. Redondi, and Matteo Cesana. Walk this way! an IoT-based urban routing system for smart cities. *Computer Networks*, 162:106857, 2019. doi:10.1016/j.comnet.2019.07.013.
- 27 R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL: <https://www.R-project.org/>.
- 28 Martin Raifer, Rafael Troilo, Fabian Kowatsch, Michael Auer, Lukas Loos, Sabrina Marx, Katharina Przybill, Sascha Fendrich, Franz-Benjamin Mocnik, and Alexander Zipf. OSHDB: a framework for spatio-temporal analysis of OpenStreetMap history data. *Open Geospatial Data, Software and Standards*, 4(1), 2019. doi:10.1186/s40965-019-0061-3.
- 29 Karl Rehrl, Stefan Kranzinger, and Simon Gröchenig. Which quality is a route? a methodology for assessing route quality using spatio-temporal metrics. *Transactions in GIS*, 25(2):869–896, 2021. doi:10.1111/tgis.12705.
- 30 Daniel A. Rodríguez, Louis Merlin, Carlo G. Prato, Terry L. Conway, Deborah Cohen, John P. Elder, Kelly R. Evenson, Thomas L. McKenzie, Julie L. Pickrel, and Sara Veblen-Mortenson. Influence of the built environment on pedestrian route choices of adolescent girls. *Environment and Behavior*, 47(4):359–394, 2015. doi:10.1177/0013916513520004.
- 31 Sophie Schetke, Salman Qureshi, Sven Lautenbach, and Nadja Kabisch. What determines the use of urban green spaces in highly urbanized areas? – Examples from two fast growing Asian cities. *Urban Forestry & Urban Greening*, 16:150–159, 2016. doi:10.1016/j.ufug.2016.02.009.
- 32 Hansi Senaratne, Amin Mobasheri, Ahmed Loai Ali, Cristina Capineri, and Mordechai (Muki) Haklay. A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, 31(1):139–167, 2017. doi:10.1080/13658816.2016.1189556.
- 33 P. N. Seneviratne and J. F. Morrall. Analysis of factors affecting the choice of route of pedestrians. *Transportation Planning and Technology*, 10(2):147–159, 1985. doi:10.1080/03081068508717309.
- 34 Monir H. Sharker and Hassan A. Karimi. Computing least air pollution exposure routes. *International Journal of Geographical Information Science*, 28(2):343–362, 2014. doi:10.1080/13658816.2013.841317.
- 35 P. Sitaraya, Y. Wang, Y. Zhang, S. Wakamiya, P. Jeszenszky, Y. Kawai, and A. Jatowt. Beyond the shortest route: A survey on quality-aware route navigation for pedestrians. *IEEE Access*, 8:135569–135590, 2020. Conference Name: IEEE Access. doi:10.1109/ACCESS.2020.3011924.
- 36 Duncan Temple Lang. *RCurl: General Network (HTTP/FTP/...) Client Interface for R*, 2021. R package version 1.98-1.3. URL: <https://CRAN.R-project.org/package=RCurl>.
- 37 Heike Tost, Markus Reichert, Urs Braun, Iris Reinhard, Robin Peters, Sven Lautenbach, Andreas Hoell, Emanuel Schwarz, Ulrich Ebner-Priemer, Alexander Zipf, and Andreas Meyer-Lindenberg. Neural correlates of individual differences in affective benefit of real-life urban green space exposure. *Nature Neuroscience*, 2019. doi:10.1038/s41593-019-0451-y.

- 38 Asha Weinstein Agrawal, Marc Schlossberg, and Katja Irvin. How far, by which route and why? a spatial analysis of pedestrian preference. *Journal of Urban Design*, 13(1):81–98, 2008. doi:10.1080/13574800701804074.
- 39 Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. URL: <https://ggplot2.tidyverse.org>.
- 40 Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi:10.21105/joss.01686.
- 41 Bin Zou, Shenxin Li, Zhong Zheng, Benjamin F. Zhan, Zhonglin Yang, and Neng Wan. Healthier routes planning: A new method and online implementation for minimizing air pollution exposure risk. *Computers, Environment and Urban Systems*, 80:101456, 2020. doi:10.1016/j.compenvurbsys.2019.101456.



# A Clustering-Based Framework for Individual Travel Behaviour Change Detection

Ye Hong ✉ 

Institute of Cartography and Geoinformation, ETH Zurich, Switzerland

Yanan Xin ✉ 

Institute of Cartography and Geoinformation, ETH Zurich, Switzerland

Henry Martin ✉ 

Institute of Cartography and Geoinformation, ETH Zurich, Switzerland

Institute of Advanced Research in Artificial Intelligence (IARAI), Austria

Dominik Bucher ✉ 

Institute of Cartography and Geoinformation, ETH Zurich, Switzerland

Martin Raubal ✉ 

Institute of Cartography and Geoinformation, ETH Zurich, Switzerland

---

## Abstract

The emergence of passively and continuously recorded movement data offers new opportunities to study the long-term change of individual travel behaviour from data-driven perspectives. This study proposes a clustering-based framework to identify travel behaviour patterns and detect potential change periods on the individual level. First, we extract important trips that depict individual characteristic movement. Then, considering trip mode, trip distance, and trip duration as travel behaviour dimensions, we measure the similarities of trips and group them into clusters using hierarchical clustering. The trip clusters represent dimensions of travel behaviours, and the change of their relative proportions over time reflect the development of travel preferences. We use two different methods to detect changes in travel behaviour patterns: the Herfindahl-Hirschman index-based method and the sliding window-based method. The framework is tested using data from a large-scale longitudinal GPS tracking data study in which participants had access to a Mobility-as-a-Service (MaaS) offer. The methods successfully identify significant travel behaviour changes for users. Moreover, we analyse the impact of the MaaS offer on individual travel behaviours with the obtained change information. The proposed framework for behaviour change detection provides valuable insights for travel demand management and evaluating people's reactions to sustainable mobility options.

**2012 ACM Subject Classification** Information systems → Geographic information systems; Information systems → Clustering; Applied computing → Transportation

**Keywords and phrases** Human mobility, Travel behaviour, Change detection, Trip clustering

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.4

### Supplementary Material

*Software (Source Code):* <https://github.com/mie-lab/change-detection>

archived at `swb:1:dir:995567f0d4599413f583b7b684a5ca7377cf7c54`

**Funding** This research was supported by the Swiss Data Science Center (SDSC).

## 1 Introduction

Individual mobility is currently primarily based on the private car. Owning a car is associated with a high level of comfort and flexibility [39], as it is always available and can reach most places while offering a safe and personal space. However, conventional car dependence is inherently unsustainable as internal combustion engine cars are a major emitter of greenhouse gases [34]. Therefore, a successful transition towards a sustainable transportation system must find ways to reduce individual car ownership and support individuals in engaging



© Ye Hong, Yanan Xin, Henry Martin, Dominik Bucher, and Martin Raubal; licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 4; pp. 4:1–4:15

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 4:2 Travel Behaviour Change Detection

in a more sustainable mobility lifestyle [42]. One of the main strategies to reduce private car ownership is Mobility-as-a-Service (MaaS), a concept where multiple shared modes are integrated with public transport to facilitate intermodal travel [31]. Despite the popularity of MaaS as a concept, there is currently only limited empirical evidence on how exactly MaaS will influence travel behaviour [16]. To evaluate the impact of MaaS, it will be necessary to collect substantial behavioural data and detect whether and to what extent individuals change their behaviour when exposed to a MaaS.

Travel behaviour refers to the decision-making process of individuals moving across space and making use of the transport facilities [12, 23]. Each individual has a set of preferred travel behaviours, collectively forming a travel pattern that is stable over the short term and evolves over the long term [9]. To correctly evaluate MaaS as an instrument for behaviour change, it is necessary to detect the long-term evolution of individuals' travel behaviour patterns. Apart from the evaluation of MaaS, detecting behaviour change on an aggregated level helps transportation planners to manage travel demand, understand the impact of transportation policies (e.g., congestion pricing, road space rationing), and evaluate people's reactions to new transportation infrastructures [27]. At the individual service level, detecting changes in travel patterns allows personalized location based services to adapt to individual travel behaviour change [17], monitor mobile phone intrusions [36], and detect hidden drivers for usage-based insurance pricing [8].

Motivated by the broad applications, many researchers have focused on travel behaviour change over the recent years [15, 20]. Studies on individual travel behaviour change mainly employed travel panel data and analysed the causes and directions of the behaviour change. For example, Lin *et al.* [24] investigated the role of social network and social environment in the relationship between residential relocation and travel behaviour change. In their study, travel behaviours are represented by trip frequency, travel time, and modal split. A study by Jain *et al.* [18] analysed the process of travel behaviour change associated with the adoption of a car-sharing service, in which the travel behaviours are represented by the amount of travel and mode choices. Overall, limited studies have focused on the detection and changing speed of the new behaviours using panel data, partially since those discrete-time panel data were not suitable for observing a dynamic process [22]. In the era of ubiquitous computing, billions of personal devices connect us to the web and enable the generation of data sets that reflect large-scale human digital traces. Compared to actively obtained travel survey data, these data sets passively and continuously record the whereabouts of individuals over time, offering new opportunities for data-driven approaches to study individual travel behaviour change.

Analysing travel behaviour and detecting its change is challenging because continuous individual movement traces are noisy, and travel patterns are latent in the movement data. To partially mitigate this issue, previous studies most often use aggregated indexes (e.g., total travel time per day) for describing travel behaviours [20]. Although this processing method can extract the essential travel behaviour, an aggregation level needs to be predefined, and some fine-grained details might be omitted during the aggregation process. Therefore, this study uses the amount of travel (trip duration and distance) and mode of transport as features to describe travel behaviours and employ clustering methods to identify travel behaviours directly at the trip level.

We develop a clustering-based framework that utilises passively tracked data to detect changes in individual travel behaviour patterns. Two change detection methods are proposed and analysed using a real-world, large-scale GPS tracking data set that evaluates the effect of introducing a MaaS offer to the study participants. The remainder of the paper is organised

as follows. Section 2 reviews related work on individual travel behaviour change detection. Section 3 presents the clustering-based change detection framework developed in our study. Section 4 describes our case study data and pre-processing steps. Section 5 explains the case study results. Section 6 summarises the main contributions of our study and highlights future research directions.

## 2 Related work

Research on individual travel behaviour change has primarily focused on short-term changes, often called anomalies, outliers, or intrapersonal variability. A generic technique used to detect this type of change when the change is not known a priori is to model the dominant pattern from the data as a standard pattern and identify observations that deviate from this pattern as potential anomalies [7]. Many methods have been developed to model the dominant pattern and detect short-term travel behaviour changes, such as clustering-based methods, frequency pattern mining methods, and generative models. An example of using a cluster-based method is given in [41]. The authors used a hierarchical clustering method to identify normal clusters of trajectories and detect anomalous taxi routing patterns that lie outside these clusters for inferring taxi fraud or traffic incidents. As a complementary example, a study by Sun *et al.* [36] utilized a frequent pattern mining technique and modelled the mobility sequence of an individual as a mobility trie. The frequent transition pattern between places is generated based on this mobility trie, and a less frequent travel sequence is considered abnormal. An example of using generative models is shown in [37], where a generative two-dimensional Latent Dirichlet Allocation model was developed to capture routine patterns of individuals and predict future movements. Trajectories with low predictability are considered abnormal and are used to indicate potential changes in travel behaviour patterns.

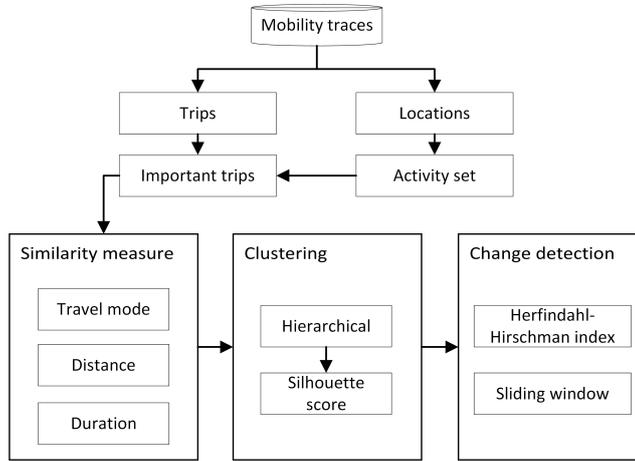
In contrast to a large number of studies in short-term pattern change detection, less attention has been given to detecting long-term, persistent pattern changes of individual travel behaviours. This long-term pattern change is also called structural pattern change, change point detection, or concept drift in time series analysis and has been studied in fields such as statistics [25, 35], econometrics [4], and sequential pattern mining [40]. However, those methods have not been well utilized in analysing the evolution of individual travel patterns.

Recently, a few studies explored the long-term travel pattern change of individuals. Zhao *et al.* [44, p. 74] defined this type of change as “abrupt, substantial, and persistent changes in the underlying patterns of travel behavior”. To detect this pattern change, they used a Bayesian approach to model the probability of a pattern change at any given time. The study examined the changes in three dimensions of travel behaviour: travel frequency, spatial dimension, and temporal dimension. Although the method is shown to be robust to noisy observations of travel behaviours, it assumes that travel incidences are independently generated from an underlying distribution in each dimension, thus missing the temporal dependency among one dimension and the correlation between different dimensions. To account for the pattern change reflected in other dimensions of travel behaviours, Jonietz and Bucher [20] considered multiple daily and weekly aggregated mobility features, including travel duration, distance, speed, CO<sub>2</sub> emission, and frequently visited places. The detection of pattern change is based on whether each feature value deviates from the historical average value by a certain threshold. The framework could signal anomalies in each feature dimension but failed to consider travel behaviour as a whole. By contrast, the clustering method used

in our study considers the correlations between different dimensions of travel behaviours (i.e., transport mode, distance, and duration) and separates them into clusters that represent different latent travel behaviour patterns.

### 3 Method

We aim to identify personal mobility preferences and detect possible travel behaviour changes over a long time scale [33]. This proposed framework consists of three main steps: (1) Using the individual conducted trips and visited locations, we define the activity set containing preferred locations and extract important trips that depict individual characteristic movement. (2) These trips are analysed to infer personal travel behaviours based on similarity measures and a clustering algorithm. (3) We then detect changes in the travel behaviours using the Herfindahl-Hirschman Index (HHI) and sliding window-based change detection methods. The flowchart of this framework is shown in Figure 1.



■ **Figure 1** The flowchart of the framework. We extract important trips based on the activity set, identify individual travel behaviours and detect changes over time via a clustering framework.

#### 3.1 Important trip extraction

Individual travel sequences are dynamic and complex [19] and often exhibit substantial variability regardless of changes in the travel pattern [44]. Therefore, the extraction of characteristic movements could benefit the travel pattern identification. From an activity-based analysis point-of-view, trips are seen as an induced demand for out-of-home activities, and trip and activity should be combined during analysis [33]. A recent study measured the location importance by its activity duration and proposed a concept of activity set, containing the most important locations in one’s daily life [2]. Following up on this idea, we identify trips that arrive at one of the locations in the activity set, noted as important trips that reflect individuals’ major travel patterns.

The activity set  $AS_i(t) = \{l_1, l_2, \dots, l_k, \dots, l_C\}$  is defined as the set of all locations  $l_k$  that the individual  $i$  visited at least twice and spent on average more than 10 minutes/week during a given time window  $\Delta t$  preceding time  $t$  [2].  $C \in \mathbb{N}$  is the cardinality of the activity set and called capacity. The activity time criterion ensures that only long-stayed locations are included in the activity set, and the time window  $\Delta t$  controls the strength of this criterion.

We further define the important trip set  $IT_i(t)$  of the individual  $i$  in week  $t$  as all trips that arrive at any location within the activity set. We set the time window  $\Delta t = 5$  weeks in this study, following an empirical study that reported that the destination-choice preferences of an individual stabilizes after five to ten weeks [33].

### 3.2 Trip similarity measurement and clustering algorithm

The trips included in any important trip set  $IT_i(t)$  are considered to contain information regarding the individuals' main travel behaviours. They are then fed into a clustering-based framework for identifying groups of trips with similar travel choices. We focus on high-level semantic information attached to each trip and consider trip mode, trip distance and trip duration as features. These are essential dimensions reflecting individual travel behaviour and have been employed in various travel behaviour change studies [24, 30].

The travel mode information of trips is represented as a sequence since each trip is a combination of triplegs with a single travel mode. This property makes the travel mode comparison a sequence similarity measurement problem. Nevertheless, we are more interested in what mode exists in a trip than the order of modes for analysing travel behaviour. Therefore, we regard travel mode sequences as sets and measure their Jaccard distance. For example, given two mode sequences, we reconstruct them into set  $A = \{mode_{A1}, mode_{A2}, \dots, mode_{Ai}\}$  and set  $B = \{mode_{B1}, mode_{B2}, \dots, mode_{Bj}\}$ , respectively. The Jaccard distance  $d_J(A, B)$  is then represented as:

$$d_J(A, B) = 1 - J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

where  $J(A, B)$  is the Jaccard similarity coefficient, defined as taking the ratio of the intersection over the union of the sets  $A$  and  $B$ . Thus, the pairwise distance matrix of trip modes  $D_{mode}$  is constructed by calculating the Jaccard distance between each pair of trips.

Trip distance and trip duration are two features characterising the amount of travel. We measure the distance and duration similarity of trips with the Euclidean distance and obtain two pairwise similarity measurement matrices  $D_{dist}$  and  $D_{dur}$ .

The selected semantic features are combined into the final similarity matrix  $D_{all}$ :

$$D_{all} = \omega_1 D_{mode} + \omega_2 D_{dist} + \omega_3 D_{dur}$$

where  $\omega_1$ ,  $\omega_2$ , and  $\omega_3$  are the corresponding weights that control the importance of  $D_{mode}$ ,  $D_{dist}$  and  $D_{dur}$ , respectively. The weight values can be defined equally for each of the dimensions (i.e.,  $\omega_1 = \omega_2 = \omega_3 = \frac{1}{3}$ ), but they can also be set differently when certain dimensions need to be strengthened. To ensure consistent distance scales, each distance matrix is *min-max* normalized to be in the range from 0 to 1.

We perform clustering on the similarity matrix to identify the group of trips with similar travel behaviour. This is achieved using hierarchical clustering that produces a hierarchy of clusters using agglomerative (bottom-up) or divisive (top-down) algorithms [21]. The linkage standard that defines how the distance is measured between two clusters is the key design choice for hierarchical clustering. Previous work reported that complete linkage is suitable for determining a relatively compact cluster [41, 1]. For complete linkage, the distance  $D(X, Y)$  between clusters  $X$  and  $Y$  is defined as:

$$D(X, Y) = \max_{x \in X, y \in Y} d(x, y)$$

where  $d(x, y)$  is the distance between element  $x \in X$  and  $y \in Y$ . To quantitatively evaluate the clustering label assignments and select the optimum cluster number, we adopt the silhouette coefficient as an internal measure of validation, which measures both the degree of cohesion within a class and the degree of dispersion between classes [32].

In this study, we adopt the complete linkage method and choose the cluster number with the highest average silhouette coefficient [43]. The outcome of this section is a class label assignment for each trip that groups similar trips to identify individuals' typical travel behaviour.

### 3.3 Travel behaviour change detection

The travel share of trip classes can show mobility choices and imply travel behaviour changes when compared across time. We denote the time series as  $X_i = [x_i(1), x_i(2), \dots, x_i(t)]$  where  $x_i(t)$  represents the trip class shares for important trips  $IT_i(t)$  at time step  $t$ . Our aim is, therefore, to detect possible changes of  $x_i(t)$  over the whole study period. We propose two such change detection methods: the HHI-based method and the sliding window-based method.

The HHI was first proposed in economics as a measure for market concentration and is currently widely used for measuring individuals' mode or activity choice variability [38, 14]. In the context of this study, HHI is adopted as a measure for the choice variability of trip classes. The HHI of trip class variability  $h_i(t)$  for individual  $i$  at time  $t$  is given as:

$$h_i(t) = \sum_{n=1}^N s_n(t)^2$$

where  $N$  is the number of different trip classes, and  $s_n(t)$  represents the trip shares of the  $n^{\text{th}}$  class conducted at time  $t$ . A higher  $h_i(t)$  value indicates that travel is concentrated on a few dominant trip classes, and a lower value suggests the different travel behaviours are more evenly selected by the individual  $i$ . We obtain  $h_i(t)$  for each timestep  $t$  and the original time series  $X_i$  is represented as a time series of HHI  $H_i = [h_i(1), h_i(2), \dots, h_i(t)]$ . As a change in  $H_i$  indicates a change in the preferences towards each trip class, we regard it as a change in individuals' travel behaviour.

To detect changes in  $H_i$ , we adopted a robust peak detection algorithm developed for time series data [5, 20]. The algorithm detects peaks in a time series when the values lie beyond a number of standard deviations from a moving average. It takes three input parameters: *lag* that controls the size of the moving window; *threshold*, denoted by  $\lambda$ , that determines the number of standard deviations (i.e., z-score); and *influence* that controls how much influence new data points will have on the moving average and standard deviation. At each time step  $t$ , a moving average  $\mu_t$  and standard deviation  $\sigma_t$  are calculated using data within the moving window. A data point is considered a peak if its value  $v > \mu_t + \lambda * \sigma_t$  or a valley if its value  $v < \mu_t - \lambda * \sigma_t$ .

The sliding window-based method processes data in a sequential fashion. Considering the time series  $X_i$ , to determine whether a change occurs from time step  $t_{start}$  to  $t_{end}$ , we measure the travel share difference of trip classes between these two time steps and compare it against a predefined threshold  $\tau$ . Operationally, we consider any trip class proportion change larger than 30% (i.e.,  $\tau = 0.3$ ) as a change in travel behaviour, in order to set a restrictive definition of change [15]. For a given time step  $t_i$ , the algorithm measures the class share difference between  $t_i$  and any time step  $t_n, n \in [1, i]$  preceding it.  $t_{start} = t_n$  is found if the difference is the largest of all possible  $n$ 's and also larger than  $\tau$ . We then

find  $t_{end} = t_m$  by maximizing the difference between  $t_{start}$  and any time step  $t_m, m \in (i, t]$  succeeding  $t_i$ . To prevent the generated change window from being too large, we impose that the share difference between  $t_{start}$  and  $t_{end}$  should be monotonically decreasing or increasing. Moreover, the algorithm ensures that no overlapping change periods for each individual are detected.

The outcome of this section is the change detection result for each individual. The HHI-based method outputs the peak detection results where sudden changes in trip class variability are recorded, whereas the sliding window-based method detects the starting and ending time steps where changes have occurred.

## 4 Case study

We adopt mobility data from a large-scale pilot study that evaluates the effect of a MaaS offer. The pilot study, conducted by the Swiss Federal Railways (SBB), is named SBB Green Class<sup>1</sup> [26, 6] (denoted as SBB GC in what follows) and involves 138 Switzerland-based participants. Although the participants were primarily selected based on their geographic location, the participation preconditions led to a bias towards the middle- and upper-class people with high mobility demand. The participants were mostly working full-time and aged  $47.3 \pm 7.6$  according to the socio-demographic survey.

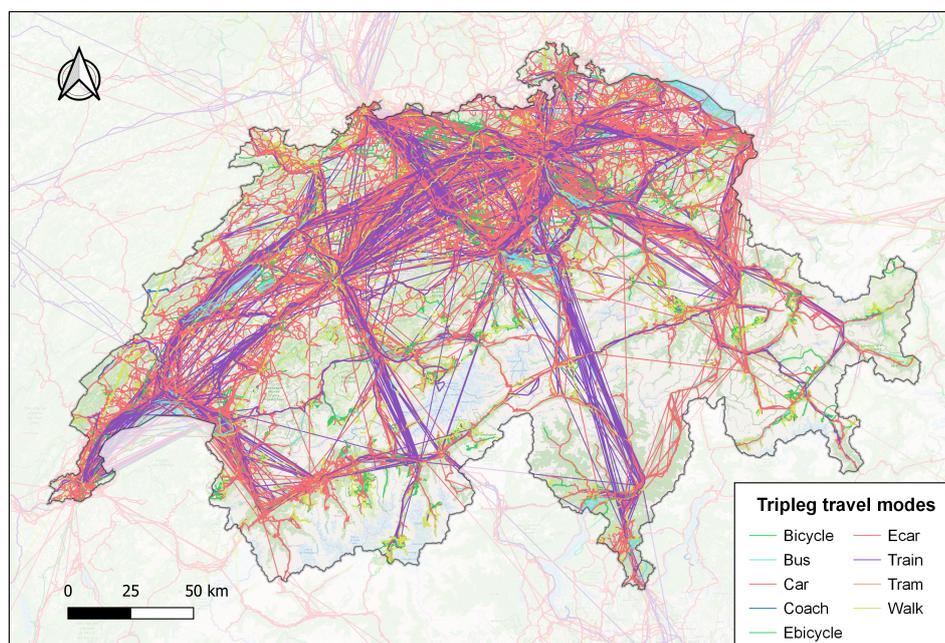
From November 2016 to January 2018, the participants were provided with a battery electric vehicle, a general public transport travel card for unlimited travel on public transport in Switzerland, as well as access to several car- and bike-sharing programs. As part of the pilot study, the participants were asked to install a GPS-tracking application on their smartphones that records their daily movement. The application uses a MOTIONTAG<sup>2</sup> back-end and segments the tracking data. It creates a *triple* when a person is moving continuously with the same mode of transport and a *staypoint* when a participant remains stationary. Study participants were required to annotate their tracking data in the application. Staypoints were annotated with a high-level purpose (home, work, errand, leisure, wait, and unknown) and triples with the used mode of transport (car, e-car, train, bus, tram, bicycle, e-bike, walk, airplane, boat, and coach). Figure 2 maps the recorded triples with user-labeled transport mode. Although we only plot the triples within Switzerland, SBB GC contains user movements all across the world - the occasionally conducted cross-border and inter-continental trips are also recorded.

The triples and staypoints provided by the GPS-tracking application are further aggregated into *trips* and *locations* according to the movement data model [3, 13]. We regard a staypoint as an *activity* if it has an important purpose (everything except for wait and unknown) or if its duration is longer than 25 minutes [26]. Trips are then constructed as the sequence of all triples between two consecutive activities. Moreover, locations are defined as important places visited more than once. Due to GPS recording error, multiple visits to the same location might create staypoints with different coordinate referencing. To tackle this problem, we use the DBSCAN method to create locations as spatially aggregated staypoints for each user. DBSCAN uses a set of neighborhood characterization parameters  $\epsilon$  and  $min\_samples$  to depict the tightness of the sample distribution [20].  $\epsilon = 50$  (m) [2] and  $min\_samples = 1$  is selected in this study, meaning that staypoints in the proximity of 50m of each other will be merged into a single location, and no staypoints will be discarded in this process.

---

<sup>1</sup> <https://bit.ly/3d0k2qD>

<sup>2</sup> <https://motion-tag.com/>



■ **Figure 2** The recorded triplets with different travel modes within Switzerland in the SBB GC data set.

We calculate the temporal tracking coverage of each user, defined as the proportion of time the user’s whereabouts are recorded in the data. To ensure high temporal tracking coverage, we only include users who are tracked for more than 300 days and whose tracking coverage is consistently higher than 60% during their tracking period. After user filtering, 193,637 staypoints and 344,740 triplets from 93 individuals remained, aggregated into 46,489 locations and 181,479 trips. These pre-processing steps are implemented using the *trackintel*<sup>3</sup> library.

We analyse the user’s moving behaviour at the trip level. Each trip’s distance and duration is measured as an aggregation of its containing triplets. Also, multiple travel modes could exist within one trip, referred to as intermodal trips [28, 29]. In fact, in the SBB GC study, 29.4% of all trips contain more than one travel mode. Walk is commonly considered a transition between other travel modes and we do not consider these trips as intermodal. As a direct result of the provided MaaS offer, mode combinations of train with car and e-car are most frequently found.

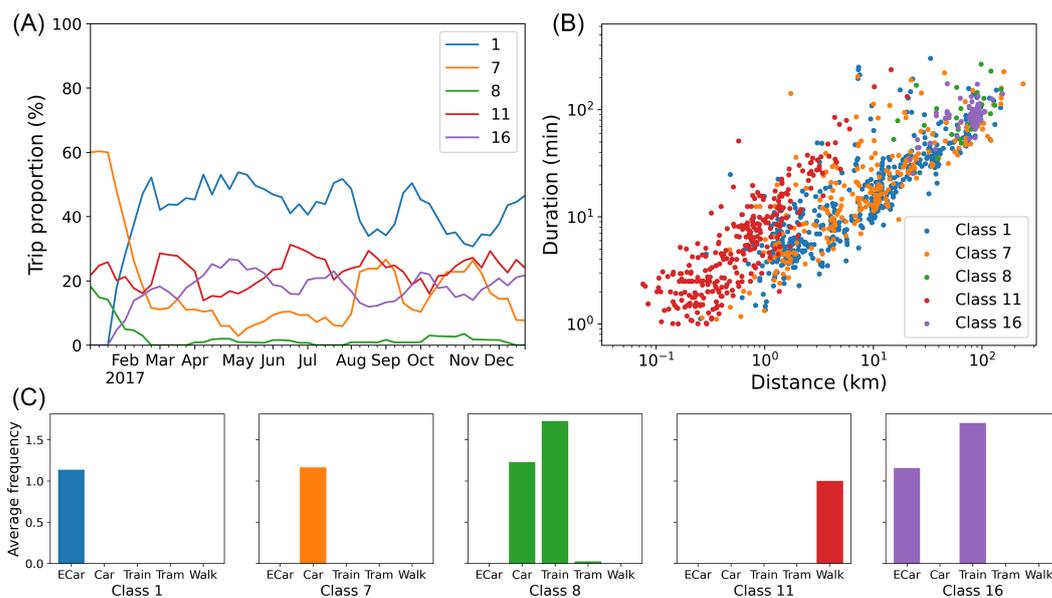
## 5 Results

### 5.1 Delineating travel behaviours

Important trips that contain an individual’s travel behaviour information are first obtained. We find that the number of important trips for each individual is stable across time; that is, individuals conduct the same number of trips to their preferred locations, despite the time of observation (detailed empirical evidence shown in Appendix A). This property makes the important trip set ideal in the study of individual travel behaviour.

<sup>3</sup> <https://github.com/mie-lab/trackintel>

The similarity measure and clustering pipeline accepts trips that belong to any important trip set, measures their semantic similarities and outputs class labels for these trips. Since we are interested in analysing travel mode change after introducing the MaaS offer and hope to interpret the change, we set the weight parameter  $\omega_1 = 0.50$  and  $\omega_2 = \omega_3 = 0.25$  in the similarity measurement. Figure 3 shows the results of the clustering label assignment of a sample user. As each trip class represents a typical travel behaviour, the proportion of travel in each trip class delineates the user travel preferences across time (Figure 3(A)). The details and separations of the travel behaviours can be visualized in each of their dimensions using distance-duration scatters (Figure 3(B)) and travel mode frequency (Figure 3(C)).



**Figure 3** Travel preference identification result for a sample user. (A) The trip proportion evolution for different trip clusters. The x-axis is the ending time of a 5-week sliding window. (B) The distance and duration scatter plot showing each cluster's distribution on a log-log scale. (C) The average frequency of travel mode within each trip cluster.

For this particular user, we observe a preference in using train and car (Class 8) and train and e-car (Class 16) mode combinations for long-distance and -duration trips. Moreover, car (Class 7) and e-car (Class 1) modes are mainly used for shorter trips, with indistinguishable duration and distance distributions. Considering the temporal dimension, we report a sharp increase in the trip proportion travelled in Class 1 and 16 at the beginning of the study period (Figure 3(A)), which is most probably due to the introduction of the MaaS offer leading the user to switch from car to e-car. Compared to frequency-based statistics, our clustering-based framework groups trips according to the similarity definition, which considers multiple trip semantic dimensions simultaneously. In short, with the information provided in Figure 3, we can delineate the individual's mobility preferences and their evolution over time.

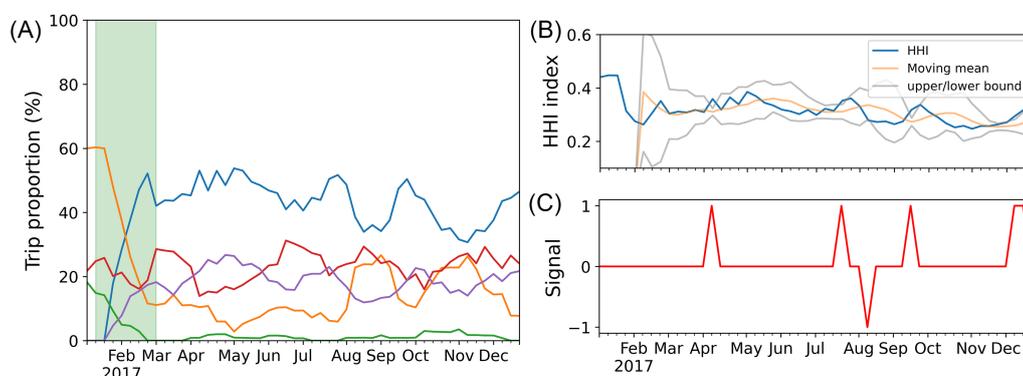
## 5.2 Travel behaviour change detection

With the proposed HHI-based and sliding window-based method, we detect change points or regions where the user has changed the travel behaviour. Figure 4 shows the change detection result for the same user as Figure 3. The sliding window-based method detects

## 4:10 Travel Behaviour Change Detection

regions where a trip proportion change larger than the threshold  $\tau$  occurs (Figure 4A). The algorithm successfully detects the travel behaviour change at the beginning of the study period. The length of the change window represents the speed of a certain change; here, the change took seven weeks. By comparing the lengths of the change windows, we can quantitatively evaluate the speed of travel behaviour change.

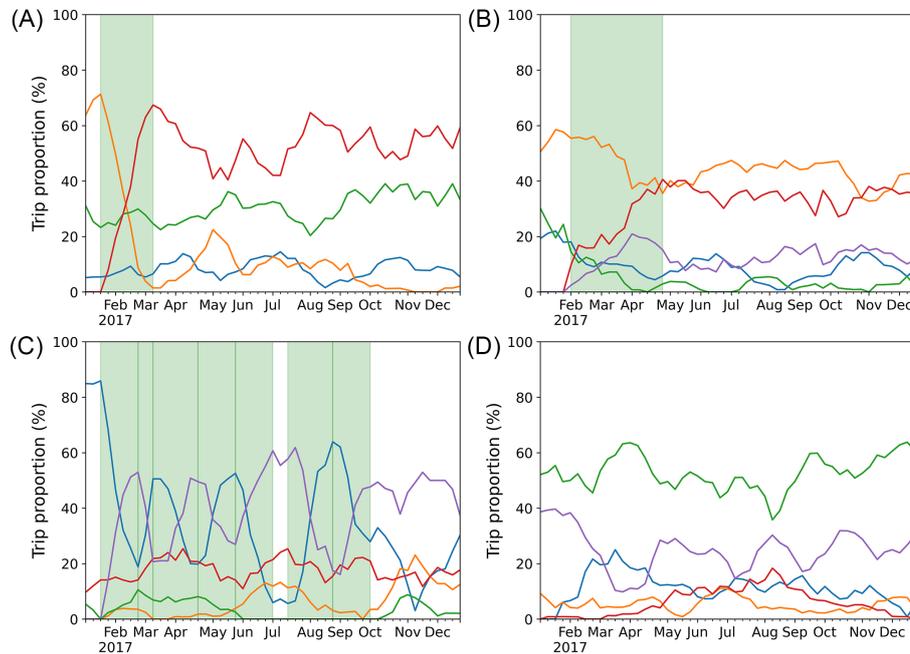
The HHI index of the trip class shares and the moving average and upper/lower bound obtained from the peak detection algorithm are shown in Figure 4B. We set the input parameter  $lag = 5$  to detect sudden changes,  $threshold = 3$  to only include extreme changing points, and  $influence = 1$  considering a change is usually substantial and persistent. The corresponding signal detection result is shown in Figure 4C. These signals capture changes in the HHI index that correspond to sudden changes in the trip class shares and individual travel behaviour. The detected signals mostly correspond to the peaks and valleys from the most dominant trip class, which also has the most considerable influence on the HHI index. However, the algorithm cannot detect signals at the beginning of the time series since the initialization of the moving time window is needed.



■ **Figure 4** The change detection results for a sample user. (A) Sliding window-based change detection. Change periods are shown in light green areas. (B) The HHI index evolution and corresponding moving mean and upper/lower bound of thresholds. (C) The peak signal detection result with peak (1) and valley (-1) signals reflect sudden changes in the HHI time series.

To validate our change detection result, we report user groups with different behaviour changes when applying the methods to individuals in the SBB GC data set. This is shown by the sliding window-based change detection result for selected users in Figure 5. In total, 92.5% of the users (86 out of 93 individuals) are observed to have started to change their travel behaviour within the first five weeks, most likely due to the MaaS offer introduction that promotes users to switch from regular cars to e-cars. However, the speed of this change varies for users, ranging from 4 to 13 weeks. With a median changing speed of 7 weeks and a standard deviation of 2.3 weeks, substantial user heterogeneity is observed in response to the same triggering event (compare Figure 5A and B as an example). This result shows that the speed of adapting to a new travel behaviour triggered by introducing new mobility options is relatively slow and heavily influenced by personal factors. As a comparison, home location change usually causes a more rapid shift in travel behaviour; an example of this is shown in Figure 5C. The individual switched back and forth between two home locations, leading to abrupt and periodic changes in the two most dominant trip classes. Also, a small number of users have no change periods detected (Figure 5D) because the trip class shares are relatively stable over the study period. From another perspective, the magnitude of the

travel behaviour change is not large enough to be detected by the proposed algorithm. This suggests that the extent of the travel behaviour change triggered by MaaS introduction also varies across individuals.



■ **Figure 5** Sliding window-based change detection result for different user types. Change periods are shown in the light green area. (A) A user who changed the travel behaviour relatively fast; (B) a user who took more time to adjust to the new travel behaviour; (C) a user who has periodic travel behaviour; and (D) a user whose travel behaviour change is not large enough to be detected by the algorithm.

## 6 Discussion and Conclusion

This study presents a clustering-based framework to detect travel behaviour changes from individual mobility traces. Specifically, we extract trips that arrive at important locations in individuals' daily mobility and consider trip mode, distance, and duration as features describing travel behaviour. These features are then fed into a trip similarity measure and clustering pipeline that generalizes individual movements into travel behaviours. Furthermore, we propose methods to detect possible changes in the travel behaviour time series. The proposed pipeline is successfully applied to a real-world GPS tracking data set collected through a pilot study in which a MaaS offer is introduced at the beginning of the tracking period.

In particular, we propose the HHI-based method and the sliding window-based method for detecting travel behaviour changes over long time series. The HHI based method analyses the variability of travel behaviour choices and detects a change if the variability significantly deviates from its previous trend. The sliding window-based method monitors the preference towards each travel behaviour and signals a change if any behaviour shows a large change in its travel share. Combined with a labelled data set, we can also attribute potential causes and analyse the detected changes. With the change duration information obtained from

the sliding window-based method, we analyse the effect of the MaaS offer on the travel behaviours in our case study. We find that these induced behaviour changes are heavily user-dependent. Moreover, compared to changes triggered by relocation, changes caused by additional mobility options usually occur slower in time.

The proposed change detection method is data-driven and identifies change periods from the statistical distribution of travel behaviour. We believe that this framework is generalisable to tracking datasets within other contexts, with flexibility given from the design choice parameters (e.g., the weights in the similarity measurement). We see several future directions based on the results of this study. First, the proposed change detection pipeline can be applied online to streaming data; however, we lack the information of whether the clustering algorithm is sensitive enough to detect newly occurred travel behaviours. This needs to be tested in an online setup. Second, a sensitivity analysis on the choice of the weight parameter for each semantic feature is beneficial if the proposed pipeline is to be employed in other applications. Third, previous research has reported that residential neighbourhood has an important impact on people's travel behaviour [10, 11]. Combining the mobility traces of SBB GC participants with land-use information could help validate the behaviour change detection results with other studies. Last, the user-dependent influence of MaaS suggests that an analysis on the relationship between individual factors and travel behaviour change is needed to understand the effect of MaaS offers on personal travel behaviour.

---

## References

- 1 Laura Alessandretti, Ulf Aslak, and Sune Lehmann. The scales of human mobility. *Nature*, 587(7834):402–407, 2020. doi:10.1038/s41586-020-2909-1.
- 2 Laura Alessandretti, Piotr Sapiezynski, Vedran Sekara, Sune Lehmann, and Andrea Baronchelli. Evidence for a conserved quantity in human mobility. *Nature Human Behaviour*, 2(7):485–491, 2018. doi:10.1038/s41562-018-0364-x.
- 3 Kay W Axhausen. Definition Of Movement and Activity For Transport Modelling. In *Handbook of Transport Modelling*, volume 1, pages 329–343. Elsevier, 2007. doi:10.1108/9780857245670-016.
- 4 Jushan Bai and Pierre Perron. Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22, 2003. doi:10.1002/jae.659.
- 5 J.P.G. van Brakel. Robust peak detection algorithm (using z-scores), 2019. URL: <https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data>.
- 6 Dominik Bucher, Henry Martin, Jannik Hamper, Atefeh Jaleh, Henrik Becker, Pengxiang Zhao, and Martin Raubal. Exploring Factors that Influence Individuals' Choice Between Internal Combustion Engine Cars and Electric Vehicles. *AGILE: GIScience Series*, 1:1–23, 2020. doi:10.5194/agile-giss-1-2-2020.
- 7 Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009. doi:10.1145/1541880.1541882.
- 8 Jie Chen, ZhongCheng Wu, and Jun Zhang. Driver identification based on hidden feature extraction by using adaptive nonnegativity-constrained autoencoder. *Applied Soft Computing*, 74:1–9, 2019. doi:10.1016/j.asoc.2018.09.030.
- 9 MC De Haas, CE Scheepers, LWJ Harms, and M Kroesen. Travel pattern transitions: Applying latent transition analysis within the mobility biographies framework. *Transportation Research Part A: Policy and Practice*, 107:140–151, 2018. doi:10.1016/j.tra.2017.11.007.
- 10 Jonas De Vos, Dick Ettema, and Frank Witlox. Changing travel behaviour and attitudes following a residential relocation. *Journal of Transport Geography*, 73:131–147, 2018. doi:10.1016/j.jtrangeo.2018.10.013.

- 11 Jonas De Vos and Frank Witlox. Transportation policy as spatial planning tool; reducing urban sprawl by increasing travel costs and clustering infrastructure and public transportation. *Journal of Transport Geography*, 33:117–125, 2013. doi:10.1016/j.jtrangeo.2013.09.014.
- 12 Anne Durand, Lucas Harms, Sascha Hoogendoorn-Lanser, and Toon Zijlstra. *Mobility-as-a-Service and changes in travel preferences and travel behaviour: a literature review*. KiM Netherlands Institute for Transport Policy Analysis, 2018. doi:10.13140/RG.2.2.32813.33760.
- 13 Ramaswamy Hariharan and Kentaro Toyama. Project Lachesis: Parsing and Modeling Location Histories. In *Geographic Information Science*, pages 106–124. Springer Berlin Heidelberg, 2004. doi:10.1007/978-3-540-30231-5\_8.
- 14 Eva Heinen and Kiron Chatterjee. The same mode again? An exploration of mode choice variability in Great Britain using the National Travel Survey. *Transportation Research Part A: Policy and Practice*, 78:266–282, 2015. doi:10.1016/j.tra.2015.05.015.
- 15 Eva Heinen and David Ogilvie. Variability in baseline travel behaviour as a predictor of changes in commuting by active travel, car and public transport: a natural experimental study. *Journal of transport & health*, 3(1):77–85, 2016. doi:10.1016/j.jth.2015.11.002.
- 16 David A Hensher, Corinne Mulley, Chinh Ho, Yale Wong, Göran Smith, and John D Nelson. *Understanding Mobility as a Service (MaaS): Past, Present and Future*. Elsevier, 2020. doi:10.1016/C2019-0-00508-0.
- 17 Haosheng Huang, Georg Gartner, Jukka M Krisp, Martin Raubal, and Nico Van de Weghe. Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, 12(2):63–93, 2018. doi:10.1080/17489725.2018.1508763.
- 18 Taru Jain, Marilyn Johnson, and Geoffrey Rose. Exploring the process of travel behaviour change and mobility trajectories associated with car share adoption. *Travel Behaviour and Society*, 18:117–131, 2020. doi:10.1016/j.tbs.2019.10.006.
- 19 Olle Järv, Rein Ahas, and Frank Witlox. Understanding monthly variability in human activity spaces: A twelve-month study using mobile phone call detail records. *Transportation Research Part C: Emerging Technologies*, 38:122–135, 2014. doi:10.1016/j.trc.2013.11.003.
- 20 David Jonietz and Dominik Bucher. Continuous Trajectory Pattern Mining for Mobility Behaviour Change Detection. In *Progress in Location Based Services 2018*, pages 211–230. Springer International Publishing, 2018. doi:10.1007/978-3-319-71470-7\_11.
- 21 Leonard Kaufman and Peter J Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*, volume 344. John Wiley & Sons, 2009. doi:10.1002/9780470316801.
- 22 Ryuichi Kitamura, Toshiyuki Yamamoto, and Satoshi Fujii. The effectiveness of panels in detecting changes in discrete travel behavior. *Transportation Research Part B: Methodological*, 37(2):191–206, 2003. doi:10.1016/S0965-8564(01)00036-2.
- 23 Meng Li, Mingqiao Zou, and Huiping Li. Urban Travel Behavior Study Based on Data Fusion Model. In *Data-Driven Solutions to Transportation Problems*, pages 111–135. Elsevier, 2019. doi:10.1016/B978-0-12-817026-7.00005-9.
- 24 Tao Lin, Donggen Wang, and Meng Zhou. Residential relocation and changes in travel behavior: what is the role of social context change? *Transportation Research Part A: Policy and Practice*, 111:360–374, 2018. doi:10.1016/j.tra.2018.03.015.
- 25 Robert Lund, Xiaolan L Wang, Qi Qi Lu, Jaxk Reeves, Colin Gallagher, and Yang Feng. Change-point Detection in Periodic and Autocorrelated Time Series. *Journal of Climate*, 20(20):5178–5190, 2007. doi:10.1175/JCLI4291.1.
- 26 Henry Martin, Henrik Becker, Dominik Bucher, David Jonietz, Martin Raubal, and Kay W Axhausen. Begleitstudie SBB Green Class - Abschlussbericht. *Working Paper No. 1439, Institute for Transport Planning and Systems, ETH Zürich*, 2019. doi:10.3929/ethz-b-000353337.
- 27 Famke JM Mölenberg, Jenna Panter, Alex Burdorf, and Frank J van Lenthe. A systematic review of the effect of infrastructural interventions to promote cycling: strengthening causal inference from observational data. *International Journal of Behavioral Nutrition and Physical Activity*, 16(1):1–31, 2019. doi:10.1186/s12966-019-0850-1.

- 28 Claudia Nobis. Multimodality: Facets and Causes of Sustainable Mobility Behavior. *Transportation Research Record*, 2010(1):35–44, 2007. doi:10.3141/2010-05.
- 29 Rebekka Oostendorp and Laura Gebhardt. Combining means of transport as a users' strategy to optimize traveling in an urban context: empirical results on intermodal travel behavior from a survey in Berlin. *Journal of Transport Geography*, 71:72–83, 2018. doi:10.1016/j.jtrangeo.2018.07.006.
- 30 Jan Prillwitz, Sylvia Harms, and Martin Lanzendorf. Interactions between Residential Relocations, Life Course Events, and Daily Commute Distances. *Transportation Research Record*, 2021(1):64–69, 2007. doi:10.3141/2021-08.
- 31 Daniel J Reck, David A Hensher, and Chinh Q Ho. MaaS bundle design. *Transportation Research Part A: Policy and Practice*, 141:485–501, 2020. doi:10.1016/j.tra.2020.09.021.
- 32 Peter J Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi:10.1016/0377-0427(87)90125-7.
- 33 Stefan Schönfelder and Kay W Axhausen. *Urban Rhythms and Travel Behaviour: Spatial and Temporal Phenomena of Daily Travel*. Ashgate Publishing, Ltd., 2010. doi:10.4324/9781315548715.
- 34 Liridona Sopjani, Jenny Janhager Stier, Mia Hesselgren, and Sofia Ritzén. Shared mobility services versus private car: Implications of changes in everyday life. *Journal of Cleaner Production*, 259:120845, 2020. doi:10.1016/j.jclepro.2020.120845.
- 35 Erla Sturludottir, Helga Gunnlaugsdottir, Olafur K Nielsen, and Gunnar Stefansson. Detection of a changepoint, a mean-shift accompanied with a trend change, in short time-series with autocorrelation. *Communications in Statistics - Simulation and Computation*, 46(7):5808–5818, 2017. doi:10.1080/03610918.2014.1002849.
- 36 Bo Sun, Fei Yu, Kui Wu, and Victor CM Leung. Mobility-based anomaly detection in cellular mobile networks. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 61–69, 2004. doi:10.1145/1023646.1023658.
- 37 Lijun Sun, Xinyu Chen, Zhaocheng He, and Luis F Miranda-Moreno. Routine Pattern Discovery and Anomaly Detection in Individual Travel Behavior. *Networks and Spatial Economics*, 2021. doi:10.1007/s11067-021-09542-9.
- 38 Yusak O Susilo and Kay W Axhausen. Repetitions in individual daily activity–travel–location patterns: a study using the Herfindahl–Hirschman Index. *Transportation*, 41(5):995–1011, 2014. doi:10.1007/s11116-014-9519-4.
- 39 Danique Ton, Lara-Britt Zomer, Florian Schneider, Sascha Hoogendoorn-Lanser, Dorine Duives, Oded Cats, and Serge Hoogendoorn. Latent classes of daily mobility patterns: the relationship with attitudes towards modes. *Transportation*, 47(4):1843–1866, 2020. doi:10.1007/s11116-019-09975-9.
- 40 Chieh-Yuan Tsai and Yu-Chen Shieh. A change detection method for sequential patterns. *Decision Support Systems*, 46(2):501–511, 2009. doi:10.1016/j.dss.2008.09.003.
- 41 Yulong Wang, Kun Qin, Yixiang Chen, and Pengxiang Zhao. Detecting Anomalous Trajectories and Behavior Patterns Using Hierarchical Clustering from Taxi GPS Data. *ISPRS International Journal of Geo-Information*, 7(1):25, 2018. doi:10.3390/ijgi7010025.
- 42 Paul Weiser, Simon Scheider, Dominik Bucher, Peter Kiefer, and Martin Raubal. Towards sustainable mobility behavior: research challenges for location-aware information and communication technology. *GeoInformatica*, 20(2):213–239, 2016. doi:10.1007/s10707-015-0242-x.
- 43 Yao Yao, Ye Hong, Daiqiang Wu, Yatao Zhang, and Qingfeng Guan. Estimating the effects of “community opening” policy on alleviating traffic congestion in large Chinese cities by integrating ant colony optimization and complex network analyses. *Computers, Environment and Urban Systems*, 70:163–174, 2018. doi:10.1016/j.compenvurbsys.2018.03.005.
- 44 Zhan Zhao, Haris N Koutsopoulos, and Jinhua Zhao. Detecting pattern changes in individual travel behavior: A Bayesian approach. *Transportation Research Part B: Methodological*, 112:73–88, 2018. doi:10.1016/j.trb.2018.03.017.

## A Stability of important trips

For each individual, we form the activity set at different time steps  $t$  and extract the important trip set  $IT_i(t)$  containing trips that arrive at any location included in the activity set. The size of the important trip set represents the number of trips to familiar locations, which we denote as trip capacity  $T_i$ . We are interested in the relation between  $T_i$  and  $t$ : if  $T_i$  does not depend on the time of observation (i.e., irrespective of  $t$ ),  $T_i$  is a conserved quantity over time; otherwise,  $T_i$  is not stable and might be influenced by individual factors and seasonality effects [19].

We first report that on a collective level, the average trip capacity  $\bar{T}$  does not depend on time  $t$ . This is shown using a linear fit of the form  $\bar{T} = a + b \cdot t$ , and through testing the hypothesis  $H_0 : b = 0$  under independent 2-samples  $t$ -tests. Using a time window size of 5 weeks, the angular coefficient  $b = -0.001 \pm 0.007$  is not significantly different from 0, and the hypothesis  $H_0 : b = 0$  cannot be rejected ( $p$ -value:  $0.86 > 0.05$ ). This stability is tested using the different choices of time window size  $\Delta t$ , as reported in Table 1. For each choice of  $\Delta t$ , we find no evidence for rejecting the hypothesis that the average trip capacity does not change in time.

At the individual level, we look at the net gain  $G_i(t)$  of the important trip set  $IT_i(t)$ . The net gain  $G_i(t) = A_i(t) - R_i(t)$  is defined as the difference between the number of trips that are respectively added  $A_i(t)$  and removed  $R_i(t)$  at time  $t$ . We find that for each individual, the average net gain across time  $\langle G_i \rangle$  is closer as its standard deviation  $\sigma_{G_i}$  to 0, i.e.,  $|\langle G_i \rangle| / \sigma_{G_i} < 1$ , indicating that  $\langle G_i \rangle$  is not significantly different than 0. Therefore, for the SBB GC population, the trip capacity is stable at the individual level. This result suggests that individuals always conduct the same amount of travel to their important activity locations. Moreover, this stability is measured as roughly 23 important trips per week for a typical user in the SBB GC data set, despite the choice of the window size  $\Delta t$  (Table 1).

■ **Table 1** Hypotheses testing for trip capacity with different window sizes. For every window size  $t$ , the null hypothesis  $H_0 : b = 0$  cannot be rejected ( $p$ -value  $> 0.05$ ).

Window size $\Delta t$	Intercept $a$	Slope $b$	Standard error	$p$ -value	$p$ -value $>0.05$
4	23.14	-0.002	0.007	0.78	True
5	23.13	-0.001	0.007	0.86	True
6	23.16	-0.002	0.007	0.81	True
8	23.24	-0.007	0.007	0.33	True
10	23.17	-0.009	0.007	0.23	True
15	22.92	-0.011	0.008	0.16	True
20	22.62	-0.012	0.009	0.20	True
30	22.09	-0.006	0.014	0.66	True
40	21.58	0.015	0.025	0.56	True



# Will You Take This Turn? Gaze-Based Turning Activity Recognition During Navigation

Negar Alinaghi ✉ 

Geoinformation, TU Wien, Austria

Markus Kattenbeck ✉ 

Geoinformation, TU Wien, Austria

Antonia Golab ✉

Geoinformation, TU Wien, Austria

Ioannis Giannopoulos ✉ 

Geoinformation, TU Wien, Austria

Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria

---

## Abstract

Decision making is an integral part of wayfinding and people progressively use navigation systems to facilitate this task. The primary decision, which is also the main source of navigation error, is about the turning activity, i.e., to decide either to turn left or right or continue straight forward. The fundamental step to deal with this error, before applying any preventive approaches, e.g., providing more information, or any compensatory solutions, e.g., pre-calculating alternative routes, could be to predict and recognize the potential turning activity. This paper aims to address this step by predicting the turning decision of pedestrian wayfinders, before the actual action takes place, using primarily gaze-based features. Applying Machine Learning methods, the results of the presented experiment demonstrate an overall accuracy of 91% within three seconds before arriving at a decision point. Beyond the application perspective, our findings also shed light on the cognitive processes of decision making as reflected by the wayfinder's gaze behaviour: incorporating environmental and user-related factors to the model, results in a noticeable change with respect to the importance of visual search features in turn activity recognition.

**2012 ACM Subject Classification** Computing methodologies → Activity recognition and understanding; Computing methodologies → Supervised learning by classification

**Keywords and phrases** Activity Recognition, Wayfinding, Eye Tracking, Machine Learning

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.5

## 1 Introduction

While decision making has seen considerable research interest in various fields beyond GIScience and LBS (see e.g., [45, 38, 17]), in our domain, up until now, very few attempts have been made to understand the behavioural correlates of decision making in human wayfinding (see Section 2). This is in contrast to the fact that understanding these decision making processes, which in wayfinding are primarily about turning activity, shows scientific and application perspectives: It can deepen the understanding of the cognitive aspects related to spatial decision-making and provide the grounds for engineering cognitively adequate wayfinding assistance systems, e.g., by monitoring user behaviour and detecting potentially wrong activities before they are actually performed.

The current work takes a step into this direction: we provide evidence that the decision making processes in terms of turning activity are reflected by gaze behaviour and the processes are, moreover, also highly affected by user-related and environmental factors. To this end, we report on an in-situ pedestrian wayfinding study ( $N = 52$ ) involving high precision GNSS and mobile eye tracking, and perform a turn-activity behaviour classification based on a series



© Negar Alinaghi, Markus Kattenbeck, Antonia Golab, and Ioannis Giannopoulos;  
licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 5; pp. 5:1–5:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of machine learning (ML) experiments, before the actual action of taking a turn. As there exists a vast body of literature which studies how basic eye-movement events reflect cognitive processes, we deliberately trained the classifiers based on a commonly-used set of gaze features. In addition to that, we also selected familiarity (measured as a self-report binary feature indicating whether the wayfinder is familiar or unfamiliar with the environment) as a simple example of a user feature, and number and spatial arrangement of street branches at each junction (see Section 2 for a reason) as indicators of environmental complexity. We provide evidence that wayfinders' gaze behavior (and, hence, their cognitive processes) are highly influenced by these factors while reaching a decision situation.

## 2 Related Work

Gaze has raised particular interest in research on cognitive load and wayfinding assistance alike. Given the aforementioned research goals, we review apt prior evidence relating to, first, gaze behaviour, cognitive load and (spatial) decision making. Second, we touch on wayfinding studies which look into gaze behaviour during this spatial activity. Third, prior work on using machine learning methods for activity classification based on eye tracking data is reviewed in order to gain an insight which classification techniques have shown promising results.

### 2.1 Gaze Analysis in Decision Making and Beyond

Since Yarbus' seminal work [55] in the 1960s, studying gaze behaviour and understanding how it reflects cognitive processes has drawn significant attention by researchers (see e.g., [22] for an early example, revealing that different cognitive tasks yield specific fixation patterns). In particular, the examination of eye movements in decision making and its underlying models has seen a long-lasting research interest (see [35] for an overview). So far, however, it remains undecided whether gaze behaviour triggers or reflects an actual choice which has been made before, and choices according to decision theory are frequently not well reflected in gaze behaviour (see [46, p. 117] for both claims). Having said this, existing literature points towards the fact that gaze bias is a good indicator for the decision to be made: A bias in gaze behavior can be observed approximately the last five seconds before we make a choice [17] and this effect is commonly referred to as the *Gaze Cascade Effect* [45]. The impact of this effect has also been studied for spatial decision making: Wiener and colleagues [53], e.g., used an eye-tracking experiment investigating the relation of gaze behavior, spatial decision making and route learning strategies, and demonstrated the importance of visual decision making in wayfinding performance. In another study, Wiener and colleagues [54] report on four studies, in which eye tracking was used to understand spatial decision making during wayfinding. Their results suggest that gaze and fixations depend on both, the task (either route learning or spatial exploration) and the environmental features.

Beyond the exploitation of fixations and saccades, however, scholars have also used pupil dilation and spontaneous eye blink rate (see [10]) as higher frequency eye tracking devices have become increasingly widespread. Numerous studies have, thereby, linked pupillometry to cognitive effort since Hess and Polt (see [20] according to [33]) found the link between higher-order cognitive processes and pupil features. Since we use a mobile eye-tracking device in our in-situ wayfinding study, we will not use any pupil-based features in our turn-activity classification task due to the frequency of 200Hz. Indeed, we will focus on a subset of fixation- and saccade-related features suggested in the literature in order to gain an insight whether real-time classification may come into range. Similarly, we have deliberately chosen not to exploit the gaze bias due to the sensory complexity this would have yielded.

## 2.2 Studying Gaze in Research on Wayfinding

Gaze has seen increased interest in recent years (see [26, p. 2–9] for an overview) with respect to both, research on wayfinding and gaze-based wayfinding assistance systems (see [14] for a dissertation spanning both aspects). Scholars have, thereby, studied gaze in wayfinding with respect to both, real-world (including indoor) and VR environments. Schnitzler and colleagues [43], for example, studied three different wayfinding aids (signage, paper or digital map) for indoor environments. By segmenting the route into 28 segments, they found, e.g., evidence for an increased visual attention at crucial decision points along a route, i.e., those which are important for floor changes. Comparing gaze behaviour during a real-world and VR wayfinding studies, Dong and colleagues [8] recently provide evidence for both, differences and commonalities of gaze behaviour between both conditions during a map reading task. For example, while the real-world condition yielded higher fixation times on the map, more fixation time was spent on self-localization in the first-person view in VR; both conditions, however, show similar fixation frequencies. In [52] differences in gaze behaviour based on a real-world wayfinding study are reported. Participants learned part of a route by incidental learning, whereas the remaining part was learned intentionally. Regardless of the learning mode, the visual salience of landmarks was a good predictor for fixation time. Intentional and incidental learning, however, show difference with respect to the structural salience of landmarks (longer fixations in case of intentional learning). [50] studies how gaze behaviour reflects subjective risk perception in bicycle riders by considering fixation duration, sight vector length and gaze angle. The results revealed a positive correlation of familiarity with the environment and cycling experience and fixation duration, gaze angle and sight vectors. Very recently, Liu and colleagues [30] study the impact of an important environmental feature, namely the regularity of road patterns, on the duration of fixations in general, mean duration, the duration of the first fixation and the fixation count. Depending on the task (shortest route selection or relative direction of destination), participants' gaze was more impacted by the (ir)regularity of the road pattern across tasks than by signs or other visual stimuli attached to buildings, a finding which is in line with the notion of an increase in wayfinding decision situations (see [16]).

## 2.3 Machine Learning for Gaze Data

Having a look at which machine learning techniques have been used to conduct classification tasks based on eye tracking, the use of Support Vector Machines (SVMs) is prevalent across different eye tracking technologies and usage scenarios. Using electrooculography to monitor eye movements, Bulling and colleagues [4], e.g., trained an SVM classifier based on a very small sample of participants ( $N = 8$ ) and achieved reasonable results regarding both, precision and recall, for an activity classification task including five everyday tasks. Similar findings for the classification of everyday tasks were reported by [44] combining gaze and motion features. Using 229 saccadic and fixation-related features, Kiefer and colleagues [25] provide evidence that six different activities which are commonly performed on maps (e.g., comparing the size of polygons when comparing e.g., lakes, or following a line representing a road) can be distinguished using a SVM classifier (achieved accuracy 78%). Liao and colleagues [29] use a Random Forest classifier and compare its performance to six different classifiers for classifying tasks according to Downs and Stea [9, pp. 125–135], based on a dataset collected during a real-world study. They use saccadic- and fixation-related features and find an accuracy of 67% when using a time window of 17 seconds.

Based on this evidence we conclude, first, to use features which are hand-crafted from eye movements like all of the aforementioned studies do and deliberately choose not to use deep learning methods. Second, regarding the reports on SVM and Random Forest algorithms in earlier studies, as well as our own pilot-test findings (see Section 4.2), we base our machine learning experiments on tree-based techniques. Using these approaches is also in line with other machine learning literature which suggests that algorithms which split the feature space are considered among the best models when it comes to small-to-medium sized structured or tabular data [48].

### 3 Data Collection, Pre-processing and Feature Extraction

Based on our goal of turning activity recognition, this section provides information on data collection and all data pre-processing steps, including the extraction of gaze features. We start with a short description of the human-subject in-situ study as we use a dataset which was collected in order to address several research problems, next to the turn activity recognition. We, then, move on to explain the synchronization of the different sensory data used. Finally, we provide details on the segmentation of the gaze data and the extraction of saccade- and fixation-related features.

#### 3.1 Data Collection

The data used in this paper is part of a larger data collection effort<sup>1</sup> addressing human spatial behavior in real-world wayfinding scenarios<sup>2</sup>. This goal renders it also a valuable source for answering other questions regarding decision-making in navigation. The data collection, which took place in 2020, required participants to walk two routes (length ranging from 0.9km to 1.3km), one of which was located in an area they were familiar<sup>3</sup> with, whereas they were unfamiliar with the other. They had to find their way by means of auditory, landmark-based<sup>4</sup>, turn-by-turn route instructions, which were provided to them on demand and as many times as they requested. We tracked participant's behaviour using a mobile eye-tracking device (PupilLabs Invisible) recording gaze positions at 200Hz and a high precision GNSS receiver (PPM 10-xx38) tracking their position in time. In total,  $N = 52$  people (27 female and 25 males,  $M(age) = 26$  years,  $SD(age) = 8.3$ ) participated in the outdoor experiments resulting in  $N = 104$  trials out of which  $N = 86$  were contained for further processing<sup>5</sup>.

#### 3.2 Data Pre-processing

To obtain the gaze-related data at each decision-point, several steps had to be taken. First, junctions had to be matched to the GPS tracks obtained in order to be able to align eye-tracking and GPS data. Subsequently, the point in time when a junction can be perceived as junction had to be approximated.

<sup>1</sup> This experiment has been described for the first time in [19] and here we only present the parts we used in our analysis.

<sup>2</sup> The data used in this paper, will be made available at <https://geo.geoinfo.tuwien.ac.at/resources/> (DOI: 10.5281/zenodo.4298703).

<sup>3</sup> During an online study participants indicated areas and places therein, with which they are familiar and provided a route connecting two of these places.

<sup>4</sup> Points of Interest were used as landmarks and chosen according to the algorithm described in [40].

<sup>5</sup> Trials had to be excluded, for example due to equipment malfunction or non-cooperative participant behaviour.

### 3.2.1 Matching Junctions to GPS Tracks

In order to match junctions to GPS trajectories, we make extensive use of VGI data (OpenStreetMap) in a multistep procedure:

**Step 1: Retrieve intersections** We retrieve all intersections of type *car and pedestrian* or *pedestrian only* within a distance of  $30m$  from the GPS track using the *Intersections Framework* [11].

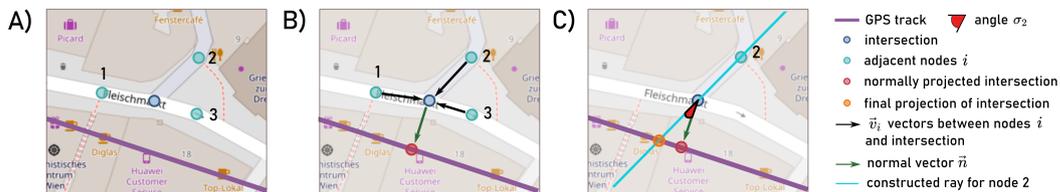
**Step 2: Filter intersections** To avoid having unreasonably short segments, we exclude any pedestrian intersections in proximity of  $50m$  to road intersections. Furthermore, due to the fact that many auxiliary path segments which are not meaningful for real-world wayfinding task (e.g., ways in parks would be obstructed by barriers) exist in OSM data, two raters individually checked each of the junctions using a web map. Subsequently, they agreed on a set of rules for exclusion and checked each of the candidate-intersections during a pair review session, ultimately eliminating 695 intersections.

**Step 3: Labeling** Based on the route instruction, we assign to each of the intersections one of the classes *turn left (TL)*, *turn right (TR)* or *non-turn (NT)*.

**Step 4: GPS track recovery** Low quality GPS tracks, which were inevitable due to the urban environment, were retraced using a polyline onto which the GPS trajectory was normally projected. This projection is applied to refine the GPS points coordinates and, at the same time, preserve the timestamps of the original points.

**Step 5: Match intersections to GPS track** We matched each of the intersections along a route to their corresponding GPS point (see Figure 1) by drawing a ray oriented according to the ways meeting in the given intersection, and intersecting it with the GPS trace.

Taken together, the procedure described above enabled us to identify the GPS point of our trajectory representing the junction and, consequently, also the corresponding timestamp at which this junction point was reached. We exploited this timestamp for segmenting the eye tracking data for our turning activity recognition experiment.

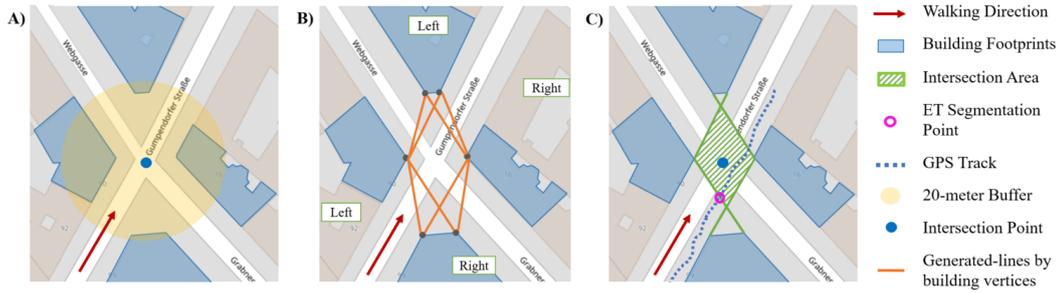


**Figure 1** Identification of GPS points corresponding to an intersection **A**: Extract adjacent nodes to which the intersection node is connected by way entities **B**: Determine the normal vector  $\vec{n}$  (green) and construct vectors  $\vec{v}_i$ , which represent the direction between each adjacent node 1,2,3 and the intersection. **C**: Construct rays passing through the adjacent nodes and the intersection. We choose the ray with the smallest angle  $\sigma_i$  between vectors  $\vec{n}$  and  $\vec{v}_i$  (in this example ray from node 2) and intersect it with the GPS track to determine the GPS position.

### 3.2.2 Eye Tracking Data Segmentation

Since we wanted to monitor gaze behaviour during decision making in real-world wayfinding, we need to identify the point in time (again based on the GPS track) at which participants were able to perceive a decision situation as such. This can only be the case where participants have a line of sight to further wayfinding options, i.e., a junction is not perceived as a point by wayfinders but as a larger area where different elements of the street network meet.

## 5:6 Will You Take This Turn?



■ **Figure 2** Extracting intersection boundary in order to have the closest point to decision-situation in reality, which is essential for eye tracking data segmentation.

Figure 2, illustrates the process for those cases in which buildings were located at a junction<sup>6</sup>: We select building footprints within a distance of  $20m$  to the intersection point for both sides of the current route segment (Figure 2 A). We, then, select the three vertices of each building polygon which are closest to the intersection point and connect each of these with the selected vertices on the opposite side of the street (the set of orange lines in Figure 2 B). Finally, we find the intersection boundary by choosing those lines from the set of orange lines which are oriented as perpendicular as possible to the given route segment (Figure 2 C). Having extracted the intersection boundaries at each junction, the closest GPS point was selected and its corresponding timestamp was derived. This point in time was found in the raw gaze-position data and flagged as the beginning of the intersection area, i.e., the point in time at which a decision would have been made already.

### 3.3 Feature Extraction

Inspired by prior evidence (see Section 2), we extracted 28 fixation-/saccade-related features (see Table 1). Since a  $200Hz$  gaze-recording frequency does not allow for velocity-based feature extractions, we used the IDT [42] as a commonly-used dispersion-based algorithm to detect fixations (gaze-dispersion threshold:  $0.02\ deg$ ; time threshold:  $100\ ms$ ) [15, 14]. All saccades were calculated based on this set of fixations. We extracted all of these features for two to ten seconds (referred to as *window* from now on) before a participant would reach the intersection area (see above). We, thereby, aim to cover the decision process as Brunye and colleagues provide evidence in [3] that ‘[w]hile intersections may prompt a decision, and elicit overt behavior that reflects a decision, the process of arriving at a spatial decision often occurs before arriving in [sic] the intersection.’

Inspired by the work on wayfinding decision situations [16], we decided to include two environmental features: namely the number of ways an intersection comprises and the skewness of these street segments. Both of these features were obtained using the *Intersection Framework* [11] and can, hence, be easily obtained on a global scale. In addition to that, we included familiarity with the environment traversed as a binary variable because prior evidence strongly suggests that familiarity has an impact on both spatial behaviour (see e.g., [50, 18, 37, 13]) and visual search behaviour (see e.g., [23, 51]). Having taken all steps for data preparation, we ended up having nine tabular datasets of dimension  $804 * 32$ .

<sup>6</sup> If no buildings were located at a junction, we found the boundary of the intersection by using a threshold of  $3.75seconds$  from the projected junction point. This resembles  $5m$  based on the assumption of an average walking speed of  $4.5km/h$  [27].

■ **Table 1** Gaze-based features extracted for turn-activity classification. For each row, the rightmost column indicates the final number of features we gained by applying the statistical measures, presented in the first column, on the gaze features represented in the second column. For instance, the first row of saccade-based features, encodes eight features, namely, mean/min/max/var of both, saccade amplitude and saccade duration.

Fixation-based Features		
mean, min, max, var frequency	duration, dispersion, dispersion X, dispersion Y -	16 1
Saccade-based Features		
mean, min, max, var skewness frequency	amplitude, duration amplitude -	8 1 1
g-l ratio (the ratio between long and short saccades)	amplitude	1

## 4 Machine Learning Experiments

### 4.1 Data Preprocessing

Prior to model training, all categorical features should be converted to numerical values, and the sample distribution across all turn-activity classes should be balanced. In order to resolve the imbalanced class issue in our dataset (“NT” class containing 636 samples, had almost four times more samples than both “TL”, containing 91 samples, and “TR”, containing 77 samples), we used Synthetic Minority Over-sampling Technique (SMOTE) for oversampling [5]. By default, SMOTE oversamples all classes to achieve equal sample sizes based on the largest sample size. Therefore, we ended up having 636 samples per class which sums up to 1908 samples in total.

### 4.2 Machine Learning Experiments

As mentioned above (see Section 3.1), we set our sights on classifying wayfinders’ turning activity, mainly by using their gaze behavior, and to do so we extracted some basic gaze-based features and prepared the data for the classification task.

During pilot testing three different models were utilized, namely, SVM-RBF, CART and Random Forest. As described in Section 2, Support Vector Machine (SVM) classifiers have been particularly popular for gaze data classification tasks. Therefore, this algorithm was implemented first for the classification task (test accuracy:  $0.58 \pm 0.05$ ). The Decision Tree (test accuracy:  $.61 \pm 0.06$ ) creates a system of rules to roughly split the feature space into several regions. Following the advantage of tree structures, the Random Forest algorithm (test accuracy:  $.77 \pm 0.06$ ) was next deployed to capture the insight of several Decision Trees in order to divide the feature space more precisely. The Random Forest algorithm yielded better test accuracy which supports the assumption that tree-based structures are more suitable for modeling the data at hand (see e.g., [48, 34, 28]). Additionally, as tree-based ensemble algorithms are reported to have multiple advantages over normal tree-based structures, including the handling of multi-collinearity in features (see e.g., [7]), we next deployed a Gradient Boosted Trees algorithm. All these classifiers were compared in the pilot testing based on achieved accuracies and their potential power to model the provided input data: Gradient Boosted Trees showed the most promising results. We, therefore, chose this class of ML algorithm to base our results on and will provide further details about this algorithm, only.

### 4.3 XGBoost Classifier

Gradient boosting is a very powerful technique for building predictive models [6]. The main idea behind this algorithm stems from the so-called “Hypothesis Boosting Problem” which states that several poor hypotheses (so called weak learners) can be converted into a very good hypothesis [24]. Compared to Random Forest classification, Gradient Boosted Trees have a lot of model capacity so they can model very complex relationships and decision boundaries. While there are various implementations of Gradient Boosted Trees available, we used XGBoost due to its scalability and high efficiency [6]. The Boosting method in XGBoost combines weak learners (i.e., trees) sequentially so that each new tree corrects the error of the previous one. We used Python 3.8 for our implementation and made use of different packages including xgboost 1.3.3 and scikit-learn 0.24.1.

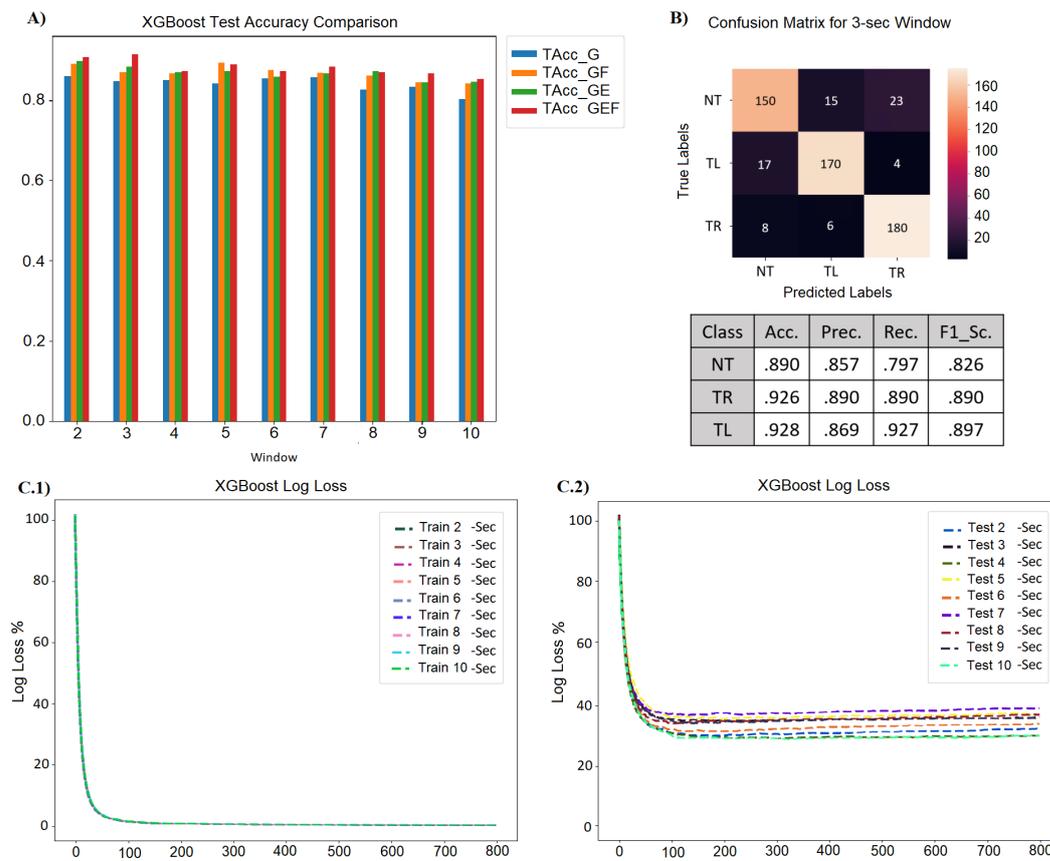
Splitting the data into 70% and 30% (train, validation) and using 20-fold validation, we performed a randomized search to tune hyper parameters. The hyper parameters performing best were: *colsample-bylevel* = 0.7, *colsample-bytree*: 0.8, *learning-rate*: 0.2, *max-depth*: 15, *min-child-weight*: 0.5, *n-estimators*: 800, *reg-lambda*: 1.0 and *subsample*: 0.5. Using the trained model, we classified the data and plotted the *mlogloss* and *merror* to check for signs of overfitting (cross entropy was used as the loss function). In addition to that, in order to avoid training and validation sessions with samples from the same user we deployed a “leave one out” method, in which all samples from one participant were left out and only used for validation. The results of our XGBoost classifier are presented in Section 5.

As mentioned above, the basic idea behind boosting algorithms is to build a weak model, making conclusions about the various feature importance and parameters, and then using those conclusions to build a new, stronger model trying to reduce the misclassification error. The most important part which allows us to come closer to an interpretation of the results is the so-called feature importance. In order to preserve both the *global* impact of features on the model and the *individualized* impact of features on a single prediction, we used the Tree SHapley Additive exPlanations (SHAP) method to calculate feature importance as it provides consistent results through handling collinearity of features automatically [31]. The interpretation of results using these features importance, is presented and discussed in Section 6.

## 5 Results

This section provides the results of our machine learning experiments for turning activity classification. We distinguish three classes, namely non-turn (NT), turn left (TL) and turn right (TR). We run our experiments based on four different sets of features: *Gaze-based* (*G*) features, *Gaze-based + Familiarity* (*GF*) features, *Gaze-based + Environmental* (*GE*) features, and *Gaze-based + Environmental + Familiarity* (*GEF*) features. As explained in Section 3.2.2, we considered these sets of features in different time windows (i.e., from two to ten second) before the actual action of turning.

The overall results in terms of *Test Accuracy* for two- to ten-seconds windows are presented in Figure 3A. Additionally, Figure 3B, presents the confusion matrix for the three-seconds window and its corresponding evaluation metrics including accuracy, precision, recall, and F1-score per class. The false positive/negative rate within turn classes (turn-left/-right) are lower than between non-turn and each of the turn classes. This observation indicates that the model has a better performance distinguishing turn left and right, than differentiating non-turn from turn classes. Figures 3C.1 and C.2, represent, as an example, the logloss error for training and test based on the GEF-dataset for all time intervals. The results show two



**Figure 3** This figure contains, **A:** Test accuracy of XGBoost classifier for four categories of features: Gaze features (TAcc-G), Gaze and Familiarity features (TAcc-GF), Gaze and Environmental features (TAcc-GE), and Gaze and Familiarity and Environmental features (TAcc-GEF); as shown in the figure adding more factors to incorporate user and environmental aspects boosts the model performance; **B:** Confusion matrix calculated for three-seconds window, and evaluation metrics including *Accuracy*, *Precision*, *Recall* and *F1-Score* per class; this indicates that the model has a better performance distinguishing left and right turns compared to non-turns from turns. **C.1** and **C.2:** Log loss plots for XGBoost classifier, respectively for *Train* and *Test* subsets, illustrating the model behavior during the training and test sessions.

important patterns: 1) the accuracy increases as the distance to the decision point decreases; 2) adding environmental and familiarity features considerably enhances model accuracy. In order to gain a deeper understanding of the results, Table 2 provides details regarding Train (TrAcc) and Test accuracies (TAcc) and the kappa coefficient for all time intervals and all combination of features. The results indicate that using XGBoost, the highest accuracy of 91.4% in turn-activity classification was achieved for the three-seconds window. Therefore, our analysis of feature importance focuses on this timeframe. Figure 4 presents the SHAP importance plots for the features in A showing the gaze features only, and B showing all three categories of features. These plots indicate that binary familiarity of participants and the skewness of the street segments at an intersection are very important for the accuracy of our classification results. A deeper understanding of the differences in importance for different feature classes can be gained from Figure 5, which shows the SHAP importance ranks (see Figure 4) for features per group of features. To visually distinguish features

## 5:10 Will You Take This Turn?

from each other, we use a combination of background colors and outline patterns: Saccadic features are depicted in yellow, fixation-related features are given in green, environmental features are shown in pink and blue denotes the familiarity feature. Count-based features are shown using a dashed outline, whereas a dotted outline marks all duration-based features and a solid outline highlights all dispersion/length-based features. This figure shows a clear pattern: Adding user-related and environmental features increases the overall number of saccadic features.

■ **Table 2** Turn activity classification results for two- to ten-seconds window (W), and the four feature sets: “Gaze” features (G), “Gaze + Familiarity” features (G+F), “Gaze + Environmental” features (G+E), “Gaze + Familiarity + Environmental” features (G+F+E).

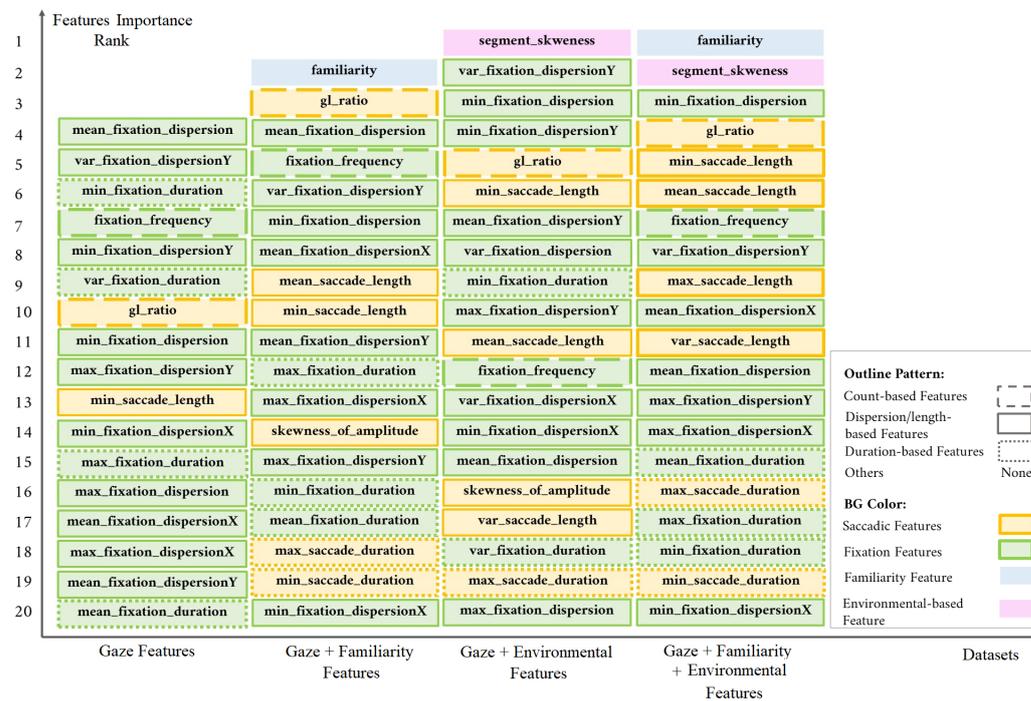
W	G			G+F			G+E			G+F+E		
	TrAcc	TAcc	Kappa	TrAcc	TAcc	Kappa	TrAcc	TAcc	Kappa	TrAcc	TAcc	Kappa
2	.842	.860	.748	.868	.891	.825	.881	.898	.772	.883	<b>.907</b>	.835
3	.872	.848	.812	.857	.869	.783	.890	.884	.741	.891	<b>.914</b>	.896
4	.853	.851	.744	.857	.867	.786	.872	.870	.788	.883	<b>.873</b>	.777
5	.838	.842	.772	.861	<b>.893</b>	.824	.876	.873	.784	.868	.890	.786
6	.827	.853	.789	.876	<b>.876</b>	.822	.898	.858	.756	.883	.873	.799
7	.823	.857	.802	.895	.868	.803	.851	.866	.830	.919	<b>.883</b>	.775
8	.815	.826	.774	.866	.861	.814	.838	<b>.872</b>	.782	.869	.870	.817
9	.850	.833	.735	.895	.845	.760	.842	.845	.705	.879	<b>.867</b>	.785
10	.847	.802	.730	.847	.841	.769	.847	.846	.716	.859	<b>.852</b>	.814



■ **Figure 4** SHAP importance plot for three-second window, derived from model, **A:** using only gaze-based features, and **B:** using all features. (Due to space restrictions, only the plot for these two feature sets are presented.)

## 6 Discussion

The results presented in Section 5, primarily, indicate that turning activity in real-world navigation scenarios using gaze-movement data and a combination of user/environmental-related features can be predicted reasonably well. It is outlined in Figure 3B, that adding



■ **Figure 5** Introducing familiarity and environmental features has a major impact on the feature importance patterns: Saccadic features which, according to the literature, reflect visual search behavior (see e.g., [36, 47]), become more dominant by encoding more information about the environment and user experience into the model.

familiarity feature increases the model accuracy, on average across all window sizes, from 83% (obtained by applying only gaze features) to 86%. Likewise, by adding environmental features to gaze data, we can see the same 3% rise in model performance from 83% (on average for all time intervals) to 86%. More importantly, adding both, user and environmental features, on top of gaze-based features boost the model performance to almost 87% on average and best performance of 91%. Taken together, these results, in general are in line with the decision-situation model proposed by Giannopoulos and colleagues [16], which state that the complexity of a wayfinding decision situation depends on user-related and environmental features, alike. The results also reiterate the findings by Brunye and colleagues [3] in two respects: Our results also suggest that the decision pattern for turning at intersections is deeply influenced by environmental elements and experience with the environment. Second, we provide evidence that the decision making in wayfinding occurs before wayfinders arrive at an intersection as, across feature sets, the highest accuracy was achieved for two to three seconds before this point in time. These findings provide the ground for discussion from both, theoretical and application aspects, and the following subsections are devoted to present them in more detail.

### 6.1 Findings on Application Level

Turning activity is one of the most important decisions made during wayfinding as almost all navigation errors occur at decision situations [1]. As a consequence, a large body of literature exists, which tries to propose solutions to either minimize the number of navigation

errors a wayfinder makes (e.g. [41]), or to minimize the impact of these errors by, for example, precalculating all possible routes close to the shortest path from the upcoming junctions [1]. A system which is capable to detect potential navigation errors would be able to pro-actively draw the wayfinders attention to this potential error and, thereby, help to avoid a wrong decision. Moreover, it has been long on debate that one of the potential drawbacks of wayfinding assistance systems, is their negative impact on humans spatial cognition and abilities (in long term) by constantly distracting users attention from the environment (see e.g. [2],[29], [39]). This detrimental effect is caused by either the need of constant visual interaction with the system, or by providing a more than necessary amount of information to the user. It has been suggested that gaze-based interactions in general can help limiting this distraction [15]. While researchers have sought for solutions to reduce the number of wayfinding instructions for years (e.g., [12]), our results allow for a personalized wayfinding assistance by monitoring gaze behavior for turning activity recognition and providing corrective instructions only if needed [32]. As an example, in case a wayfinder is supposed to continue straight ahead but shows a tendency to take a turn, the system can provide her/him with a corrective instruction; otherwise there is no need to distract the wayfinder by repeating the same instruction).

## 6.2 Findings on Theoretical Level

The findings of this paper in terms of *the role of each feature in visual search behavior during wayfinding decision situations*, can deepen our understanding of decision making in real-world wayfinding. Considering only gaze-based features (see left column of Figure 5), we observe two patterns: First, in total 15 out of 17 most important features are fixation-related. One potential explanation, which is in line with the fact that fixation-related features convey signs of attention and information processing [22], is that this dominance reflects the importance of *attention* at decision point. Prior evidence (see [3, 53]), suggests already that deciding to take a turn is a dynamic decision making process consisting of stimulus processing and action preparation which requires an attention shift. The second pattern we observe from Figure 5, relates to the equal number and importance of duration-based and dispersion-based features (note that the number of dispersion-related features are three times more than duration-based features and the same ratio holds here): Two fixation duration based features are among the 50% most important features, which is in line with the known positive correlation of fixation duration with cognitive processing and scene perception [21]. Inline with current evidence (see e.g., [22, 49]), these two patterns suggest that using only gaze information, the turn-activity decision is more reflected by features which are representative for cognitive processing and scene perception. Adding familiarity with the environment as a user-related feature (second left column of Figure 5), boosts the performance of the model and, at the same time, increases the number of saccadic features to six. Simultaneously, the importance of duration-based features (both fixation and saccade-related) slightly decreases. Saccadic features in general represent visual search activity rather than attention or concentration, and more specifically saccade amplitude, is known to reflect search task difficulty [36] and presence of high-frequency visual information [47]. This result can, hence, be interpreted as changing the importance of visual search in predicting a turn-activity decision. This would be in line with the fact that fixation behavior is particularly biased by familiarity or level of expertise [50]. Adding familiarity renders itself most important; moreover, it also redirects the model's attention towards saccadic features. This finding, though, leaves much room for further research: Arguably, familiarity somehow encapsulates most of the fixation behavior in itself, and lets the model seek for meaningful patterns in saccadic features. This

observation requires further assessment of familiarity in order to unfold the differences in gaze behavior of familiar vs unfamiliar wayfinders. Similar to adding user-related features, an increase in presence of visual search related features (i.e., saccadic features), occurs also when adding environmental factors (third left column of Figure 5). We used skewness of street segments and the number of branches an intersection has, as proxy for the difficulty of the decision situation in terms of urban configuration. Due to the collinearity of both features, only skewness is retained by the SHAP procedure. This may be an indicator for the fact that skewness is a better (i.e., more precise than the number of branches) proxy for the difficulty of the decision situation, with respect to environmental configuration. The most important pattern was revealed when we added both familiarity and environmental features (right column of Figure 5). As is shown in Figure 5, familiarity remains the most important feature, which can be explained by the fact that gaze-behavior is heavily biased by user-related factors, and particularly the level of expertise of which familiarity with the environment is an instance of. In addition to that, a slight increase in number *and* importance of saccadic features, and moreover, a sharp drop in importance for duration-based features arises. This observation may indicate that environmental and user-related features encode important aspects of the decision situation and, thereby, redirect the model's focus more to visual search behavior.

## 7 Conclusion and Future Work

In this paper we provide evidence that it is possible to classify the turning activity of wayfinders based on a combination of gaze-data, wayfinder familiarity (measured as a binary variable) and two environmental factors regarding the morphology of an intersection. To this end, we used data collected during an in-situ wayfinding experiment and compared different time windows and different classification algorithms. The highest accuracy was found for an XGBoost classifier, which achieved 91% overall accuracy in turn-activity classification for a window size of three seconds. We discussed our findings from two main perspectives: First, we shed light on the applicational prospects with respect to wayfinding assistance systems capable to detect potential navigation errors; second, we explored which gaze features are most important and provided explanations based on the existing literature. Taken together, these results raise further research questions. While we have deliberately used a basic set of gaze features combined with a single user-related and two basic environmental features, it would be interesting to see whether more sophisticated features (e.g., incorporating different levels of familiarity) convey even more information regarding the decision making process. Besides, according to our findings, adding user-related and environmental factors to gaze features, alters the pattern of fixation-related and saccadic features, in terms of the number and importance of each gaze-event category. It is, therefore, worthwhile to disentangle the relationship between these gaze features and the familiarity/environmental factors. Another interesting pattern which needs deeper exploration, relates to count-based fixation and saccadic features. So far, these have been used mostly as an indication of semantic importance and search-task difficulty [21]. Therefore, a future study can introduce semantic information of areas of interest (AOI) to the model, and observe why count-based features retain their importance in the model. Eventually, the differences in false positive/negative rates between different classes observed in the confusion matrix, may suggest missing environmental or user characteristics in the model. Therefore, a future study may for instance, incorporate head/body movements to the model to see if this behavioral source of information can lead to

better classification results. In addition to that, our findings open up several other questions regarding the general research direction of this paper:

1. When is the actual onset of the decision process for turn-activity, does this point in time vary for different phases of wayfinding (as proposed by [9]), and is this point in time stable as travel time/distance covered increases?
2. Why are some of the features more effective with respect to predicting particular classes of turn-activity? For instance, why is familiarity more effective on predicting non-turn activity as suggested by Figure 4?
3. Does the so-called cascade effect also hold for turn-activity recognition based on gaze-behaviour in real-world navigation?

---

## References

- 1 D. Amores, E. Tanin, and M. Vasardani. A proactive route planning approach to navigation errors. *Int J of Geographical Information Science*, pages 1–37, 2020.
- 2 A. Brügger, K. Richter, and S. Fabrikant. How does navigation system behavior influence human behavior? *Cognitive Research: Principles and Implications*, 4(1):5, December 2019.
- 3 T. T. Brunyé, A. L. Gardony, A. Holmes, and H. A. Taylor. Spatial decision dynamics during wayfinding: intersections prompt the decision-making process. *Cognitive Research: Principles and Implications*, 3(1), 2018.
- 4 A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster. Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):741–753, 2011.
- 5 N. V Chawla, K. W Bowyer, L. O Hall, and W P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *J of artificial intelligence research*, 16:321–357, 2002.
- 6 T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4), 2015.
- 7 T. G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- 8 W. Dong, H. Liao, B. Liu, Z. Zhan, H. Liu, L. Meng, and Y. Liu. Comparing pedestrians’ gaze behavior in desktop and in real Envs. *Cartography and Geographic Information Science*, 47(5):432–451, 2020.
- 9 R. Downs and D. Stea. *The world in the head Maps in minds: Reflections on cognitive mapping*. Harper & Row, 1977.
- 10 M. K. Eckstein, B. Guerra-Carrillo, A. T. Miller Singley, and S. A. Bunge. Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Developmental Cognitive Neuroscience*, 25:69–91, 2017.
- 11 P. Fogliaroni, D. Bucher, Nikola J., and I. Giannopoulos. Intersections of our world. In *10th Int Conf on Geographic Information Science*, 2018.
- 12 A. U. Frank. Pragmatic information content—how to measure the information in a route. *Foundations of geographic information science*, page 47, 2003.
- 13 N. Gale, R. G. Golledge, W. C. Halperin, and H. Couclelis. Exploring spatial familiarity. *Professional Geographer*, 42(3):299–313, 1990.
- 14 I. Giannopoulos. *Supporting Wayfinding Through Mobile Gaze-Based Interaction*. PhD thesis, ETH Zurich, 2016. URL: <https://www.research-collection.ethz.ch/handle/20.500.11850/116375>.
- 15 I. Giannopoulos, P. Kiefer, and M. Raubal. Gazenav: gaze-based pedestrian navigation. In *Proc of MobileHCI 2015*, pages 337–346, 2015.
- 16 I. Giannopoulos, P. Kiefer, M. Raubal, K. Richter, and T. Thrash. Wayfinding Decision Situations: A Conceptual Model and Evaluation. In *Proc of GIScience 2014*, 2014.

- 17 K. Gidlöf, A. Wallin, R. Dewhurst, and K. Holmqvist. Using eye tracking to trace a cognitive process: Gaze behaviour during decision making in a natural Env. *J of Eye Movement Research*, 6(1):1–14, 2013.
- 18 L. Gokl, M. Mc Cutchan, B. Mazurkiewicz, P. Fogliaroni, and I. Giannopoulos. Towards Urban Env Familiarity Prediction. *Advances in Cartography and GIScience of the ICA*, 2(November):1–8, 2019.
- 19 A. Golab, M. Kattenbeck, G. Sarlas, and G. Giannopoulos. It’s also about timing! when do pedestrians want to receive navigation instructions. *SPAT COGN COMPUT*, 2021. accepted for publication.
- 20 E. H Hess and J. M. Polt. Pupil size as related to interest value of visual stimuli. *Science*, 132(3423):349–350, 1960.
- 21 K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.
- 22 M. Adam Just and Patricia A. C. Eye fixations and cognitive processes. *Cognitive psychology*, 8(4):441–480, 1976.
- 23 A. Kafkas and D. Montaldi. Familiarity and recollection produce distinct eye movement, pupil and medial temporal lobe responses when memory strength is matched. *Neuropsychologia*, 50(13):3080–3093, 2012.
- 24 M. Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.
- 25 P. Kiefer, I. Giannopoulos, and M. Raubal. Using eye movements to recognize activities on cartographic maps. *Proc of SIGSPATIAL 2013*, pages 478–481, 2013.
- 26 P. Kiefer, I. Giannopoulos, M. Raubal, and A. Duchowski. Eye tracking for spatial research: Cognition, computation, challenges. *SPAT COGN COMPUT*, 17(1-2):1–19, 2017.
- 27 R. V Levine and A. Norenzayan. The pace of life in 31 countries. *J of cross-cultural psychology*, 30(2):178–205, 1999.
- 28 B. Li, L. Peng, and B. Ramadass. Accurate and efficient processor performance prediction via regression tree based modeling. *Journal of Systems Architecture*, 55(10-12):457–467, 2009.
- 29 H. Liao, W. Dong, H. Huang, G. Gartner, and H. Liu. Inferring user tasks in pedestrian navigation from eye movement data in real-world Envs. *IJGIS*, 33(4):739–763, April 2019.
- 30 B. Liu, W. Dong, Z. Zhan, S. Wang, and L. Meng. Differences in the gaze behaviours of pedestrians navigating between regular and irregular road patterns. *ISPRS Int J of Geo-Information*, 9(1), 2020.
- 31 S. Lundberg and S. Lee. A unified approach to interpreting model predictions. *Proc. of NIPS 2017*, page 4768–4777, 2017.
- 32 B Mazurkiewicz, M. Kattenbeck, and I Giannopoulos. Navigating your way! increasing the freedom of choice during wayfinding. In *11th International Conference on Geographic Information Science (GIScience 2021) - Part II*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021.
- 33 N. Venkata Kartheek Medathati, R. Desai, and J. Hillis. Towards inferring cognitive state changes from pupil size variations in real world conditions. In *Proc of ETRA 2020*, 2020.
- 34 I Nitze, U Schulthess, and H Asche. Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification. *Proc. of the 4th GEOBIA*, 35, 2012.
- 35 J. L. Orquin and S. Mueller Loose. Attention and choice: A review on eye movements in decision making. *Acta Psychologica*, 144(1):190–206, 2013.
- 36 M. H Phillips and J. A Edelman. The dependence of visual scanning performance on saccade, fixation, and perceptual metrics. *Vision research*, 48(7):926–936, 2008.
- 37 L Piccardi, M Riseti, and R Nori. Familiarity and Enval Representations of a City: A Self-Report Study. *Psychological Reports*, 109(1):309–326, August 2011.
- 38 R. Pieters and L. Warlop. Visual attention during brand choice: The impact of time pressure and task motivation. *Int J of Research in Marketing*, 16(1):1–16, 1999.

- 39 C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe. Task and context determine where you look. *J of Vision*, 7(14):1–20, 2007.
- 40 A. Rousell and A. Zipf. Towards a landmark-based pedestrian navigation service using OSM data. *ISPRS Int J of Geo-Information*, 6(3), 2017.
- 41 R. A. Ruddle, E. Volkova, B. Mohler, and H. H. Bühlhoff. The effect of landmark and body-based sensory information on route knowledge. *Memory & cognition*, 39(4):686–699, 2011.
- 42 D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proc of ETRA 2000*, page 71–78, 2000.
- 43 V. Schnitzler, I. Giannopoulos, C. Hölscher, and I. Barisic. The interplay of pedestrian navigation, wayfinding devices, and Enval features in indoor settings. In *Proc of ETRA 2016*, pages 85–93, 2016.
- 44 Y. Shiga, T. Toyama, Y. Utsumi, K. Kise, and A. Dengel. Daily activity recognition combining gaze motion and visual features. *Adjunct Proc of UbiComp 2014*, pages 1103–1111, 2014.
- 45 S. Shimojo, C. Simion, E. Shimojo, and C. Scheier. Gaze bias both reflects and influences preference. *Nature Neuroscience*, 6(12):1317–1322, 2003.
- 46 N. Stewart, F. Hermens, and W. J. Matthews. Eye Movements in Risky Choice. *J of Behavioral Decision Making*, 29(2-3):116–136, 2016.
- 47 B. W Tatler, R. J Baddeley, and B. T Vincent. The long and the short of it: Spatial statistics at fixation vary with saccade amplitude and task. *Vision research*, 46(12):1857–1862, 2006.
- 48 J. Treboux, D. Genoud, and R. Ingold. Decision tree ensemble vs. nn deep learning: efficiency comparison for a small image dataset. In *Proc. of the IWBI 2018*, pages 25–30, 2018.
- 49 P Unema. Differences in eye movements and mental work-load between experienced and inexperienced motor vehicle drivers. *Visual search*, pages 193–202, 1990.
- 50 R. von Stülpnagel. Gaze behavior during urban cycling: Effects of subjective risk perception and vista space properties. *Transportation Research Part F: Traffic Psychology and Behaviour*, 75:222–238, 2020.
- 51 Q. Wang, P. Cavanagh, and M. Green. Familiarity and pop-out in visual search. *Perception & Psychophysics*, 56(5):495–500, September 1994.
- 52 F. Wenzel, L. Hepperle, and R. von Stülpnagel. Gaze behavior during incidental and intentional navigation in an outdoor Env. *SPAT COGN COMPUT*, 17(1-2):121–142, 2017.
- 53 J. M. Wiener, O. de Condappa, and C. Hölscher. Do you have to look where you go? Gaze behaviour during spatial decision making. *Proc of CogSci 2011*, pages 1583–1588, 2011.
- 54 J. M. Wiener, C. Hölscher, S. Büchner, and L. Konieczny. Gaze behaviour during space perception and spatial decision making. *Psychological Research*, 76(6):713–729, 2012.
- 55 A. L Yarbus. *Eye movements and vision*. New York: Plenum Press, 1967.

# Bicriteria Aggregation of Polygons via Graph Cuts

Peter Rottmann ✉

Institute of Geodesy and Geoinformation, University of Bonn, Germany

Anne Driemel ✉ 

Hausdorff Center for Mathematics, University of Bonn, Germany

Herman Haverkort ✉

Institute of Computer Science, University of Bonn, Germany

Heiko Röglin ✉

Institute of Computer Science, University of Bonn, Germany

Jan-Henrik Haurert ✉ 

Institute of Geodesy and Geoinformation, University of Bonn, Germany

---

## Abstract

We present a new method for the task of detecting groups of polygons in a given geographic data set and computing a representative polygon for each group. This task is relevant in map generalization where the aim is to derive a less detailed map from a given map. Following a classical approach, we define the output polygons by merging the input polygons with a set of triangles that we select from a constrained Delaunay triangulation of the input polygons' exterior. The innovation of our method is to compute the selection of triangles by solving a bicriteria optimization problem. While on the one hand we aim at minimizing the total area of the outputs polygons, we aim on the other hand at minimizing their total perimeter. We combine these two objectives in a weighted sum and study two computational problems that naturally arise. In the first problem, the parameter that balances the two objectives is fixed and the aim is to compute a single optimal solution. In the second problem, the aim is to compute a set containing an optimal solution for every possible value of the parameter. We present efficient algorithms for these problems based on computing a minimum cut in an appropriately defined graph. Moreover, we show how the result set of the second problem can be approximated with few solutions. In an experimental evaluation, we finally show that the method is able to derive settlement areas from building footprints that are similar to reference solutions.

**2012 ACM Subject Classification** Information systems → Geographic information systems; Theory of computation → Computational geometry

**Keywords and phrases** map generalization, aggregation, graph cuts, bicriteria optimization

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.6

## 1 Introduction

Map generalization is the process of deriving a less detailed map from a given map. It comprises multiple sub-tasks such as the selection, simplification, aggregation, and displacement of objects. In this paper we address the task of detecting polygon groups and aggregating each group to a single polygon, which we simply refer to as *aggregation of polygons*. This task is relevant, e.g., to derive settlement areas from mutually disjoint building footprints.

A popular method for the aggregation of polygons is the *adopt merge amalgamation operator* proposed by Jones et al. [19], which is based on a constrained Delaunay triangulation of the space not covered by the input polygons; see Fig. 1. The approach is to select a set  $T' \subseteq T$  from the set  $T$  of triangles of the triangulation to glue together groups of input polygons. More precisely, the connected regions in the union of the triangles in  $T'$  and the input polygons constitute the output polygons. We propose a new method that follows this approach. Our focus is to find an optimal selection of triangles, whereas Jones et al. [19] left it largely open how the selection  $T'$  of triangles is computed. They rather generally



© Peter Rottmann, Anne Driemel, Herman Haverkort, Heiko Röglin, and Jan-Henrik Haurert; licensed under Creative Commons License CC-BY 4.0

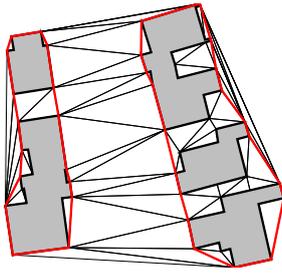
11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 6; pp. 6:1–6:16

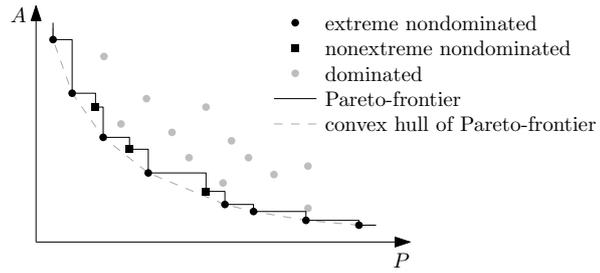
Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Input polygons (filled gray) aggregated to larger ones (red lines).



■ **Figure 2** Different types of solutions and Pareto-frontier of a bicriteria optimization problem.

recommended to use rules with thresholds. With our work we thus aim to overcome the issue raised by Li et al. [21], who noticed that the aggregation of polygons has often been discussed on a general conceptual level and that difficulties arise when engineering a method in detail. In particular, computing the set  $T'$  of triangles based on rules with thresholds requires finding the right threshold values, which is a cumbersome task.

Since setting the parameters of an algorithm is a challenge of general relevance, we do not only aim to provide a new method for polygon aggregation but also set off to develop a generic approach for the systematic exploration of different parameter values. Generally, map generalization aims at finding a good balance between (i) the preservation of the information given with the input map and (ii) the legibility of the output map [7]. We will give particular consideration to setting a parameter that balances between these two objectives.

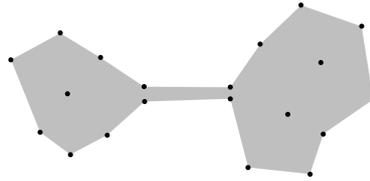
In our method for polygon aggregation, each of the two general objectives of map generalization is implemented with one basic quantitative measure. We consider the preservation of information by minimizing the total area of the output polygons, meaning that only little area should be added to the input polygons when merging the selected triangles with them. Legibility is considered by minimizing the total boundary length (or *perimeter*) of the output polygons, which can be considered as an implementation of Tufte's minimum-ink principle [38]. A parameter  $\alpha$ , which we call *balance factor*, is used to combine these objectives with a weighted sum. To formalize this, we refer with  $A(S)$  to the area and with  $P(S)$  to the perimeter of the union of all polygons in a set  $S$ , where the union can be a polygon or a multipolygon. For a single polygon  $p$  we simply refer with  $A(p)$  and  $P(p)$  to its area and perimeter, respectively. With this we state the problem we aim to solve as follows.

► **Problem 1.** *Given a subdivision of the plane as a set  $S$  of  $n$  simple polygons, a set  $B \subseteq S$ , and a balance factor  $\alpha \in [0, 1]$ , select a subset  $S'$  with  $B \subseteq S' \subseteq S$  minimizing  $f_\alpha(S') = \alpha \cdot A(S') + (1 - \alpha) \cdot P(S')$ .*

In our application,  $B$  is the set of input polygons, e.g., building footprints. The set  $S$  contains all polygons of a planar partition, including the polygons in  $B$  and the triangles of a triangulation partitioning the space not covered by the input polygons. Note that the restriction to a triangulation is not necessary, i.e., one may use any other partition of the plane instead. The requirement  $B \subseteq S' \subseteq S$  means that the input polygons have to be in the selection  $S'$ . The balance factor  $\alpha$  combines the two objectives with a weighted sum, yielding the overall objective function  $f_\alpha$ . To state that a solution to Problem 1 is optimal for a certain balance factor  $\alpha$ , we refer to it as an  $\alpha$ -optimal solution.

Finally, with the following problem we address the challenge of setting the parameter  $\alpha$ .

► **Problem 2.** *Given a subdivision of the plane as a set  $S$  of  $n$  simple polygons and a set  $B \subseteq S$ , find a set containing for every  $\alpha \in [0, 1]$  an optimal solution to Problem 1.*



■ **Figure 3** An  $\alpha$ -shape that generates a narrow bridge between two point sets.

By not requiring a pre-set value for  $\alpha$ , our method becomes parameter free. However, it now returns a set of solutions instead of a single one. This can be useful to let an expert choose from a set of alternative solutions. Moreover, the different solutions could be evaluated automatically by comparing them with a reference solution, to infer a suitable value for  $\alpha$ .

According to the definition of Problem 2, if for a fixed  $\alpha$  there are multiple optimal solutions, only one of them has to be included in the result set. This corresponds to finding what is called the set of all *extreme nondominated points*, which together with the *nonextreme nondominated points* constitute the *Pareto-frontier* [3]; see Fig. 2.

We now summarize our contribution and give an outline of the paper.

### Our Contribution

1. We show how to solve Problem 1 efficiently by computing a minimum cut in an appropriately defined graph. This approach is inspired from image analysis, where graph cuts are commonly used for image segmentation and related problems.
2. We show that it suffices to include  $O(n)$  solutions in the result set of Problem 2 and that these solutions are geometrically nested. The linear size of the result set implies that Problem 2 can be solved with  $O(n)$  min-cut computations via a generic recursive algorithm. We also show how the recursive algorithm can be used to compute a small set of solutions approximating the result set of Problem 2.
3. We evaluate our method on real data showing its applicability in practice.

In the following, we review related work (Sect. 2), present the three above-mentioned contributions (Sects. 3–5), and provide a conclusion and outlook on future work (Sect. 6).

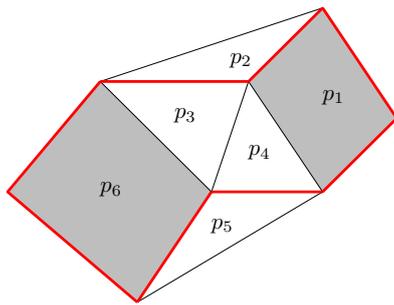
## 2 Related Work

Automatic map generalization remains a big challenge despite decades of research. The challenge lies on the one hand in the complex interplay between different processes of map generalization – a way to deal with this is to use multi-agent systems for the orchestration of multiple map generalization operators [15, 23]. On the other hand, the challenge lies in the acquisition of cartographic knowledge in a form that can be used by a computer. Machine learning approaches have been proposed to solve this task, with a recent shift towards deep learning [14, 37]. While we focus on a single map generalization operator, the aggregation of polygons, our method for exploring different parameter values can be used for automatic parameter tuning and, thus, considered as a machine-learning contribution. Polygon aggregation is relevant when generalizing categorical coverage maps [16, 18] or choropleth maps [27], where the polygons form a mosaic. These tasks are similar to districting tasks where the aim is to group small areal units to form larger regions such as electoral districts or police districts [10, 20]. In this paper, however, we deal with the aggregation of potentially non-adjacent polygons, e.g., buildings.

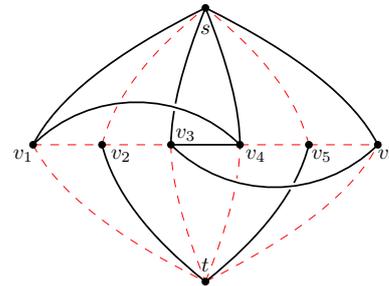
The aggregation of polygons is closely related to the aggregation of points. The common goal is to find a single polygon or multiple polygons enclosing the input data. A naïve method is to compute the convex hull of all input points. However, since this may enclose large empty regions, generalizations of the convex hull such as  $\alpha$ -shapes [13] have been developed. As an unwanted side effect,  $\alpha$ -shapes tend to introduce narrow bridges between two nearby point sets; see Fig. 3. Such a bridge can consist of a single edge, in which case it can be easily removed [4], but handling bridges of multiple parallel edges is not straight forward. Similar issues can arise with the concave hull introduced Moreira and Santos [26], which is based on  $k$ -nearest neighbours clustering. In contrast, our method would never connect two polygons with a narrow bridge as the one in Fig. 3 since *not* selecting the bridge would yield a solution with both a smaller perimeter and less area. Note that the parameter  $\alpha$  of our method is different from that of  $\alpha$ -shapes. Duckham et al. [12] defined the  $\chi$ (chi)-hull as a further generalization of the convex hull. First, all points are triangulated using a Delaunay triangulation. Then, all boundary edges that are longer than a threshold are removed. This procedure always returns a single polygon without holes. This may include large empty regions and does not separate groups of points from each other. To address the latter issue, Duckham et al. [12] suggest to identify clusters in a pre-processing step. However, this does not prevent the method from covering large empty regions within a cluster.

With respect to the aggregation of polygons, Jones et al. [19] proposed a method for merging polygons by selecting triangles of a constrained Delaunay triangulation, which they call *adopt merge amalgamation*. They do not specify the criteria for the selection of the triangles but generally recommend to use rules based on thresholds on the triangles' edge lengths. Using such rules the adopt merge amalgamation operator has been implemented and experimentally evaluated by Li and Ai [22]. They showed that the method tends to generate narrow bridges that can consist of a single triangle touching an input polygon with only one of the triangle's vertices. The method of Li et al. [21] overcomes this deficit by selecting sequences of triangles instead of single ones. However, the rules used to govern the selection are set up to aggregate polygons with parallel boundaries separated by long and narrow corridors. With this the method can be used to derive built-up areas from city blocks but not, e.g., settlement areas from footprints of detached houses. Sayidov et al. [33] compute groups of polygons using a triangulation-based method and suggest to compute a representative polygon for each group in a separate processing step. To accomplish this step, automatic typification method can be used [1, 6, 24]. Similarly, Steinger et al. [36] suggest a method that first detects groups of islands and then generates a representative polygon for each group. They define the groups with a subgraph of a minimum spanning tree and generate for each group the convex hull. Damen et al. [9] present an approach for building generalization based on morphological operators. They combine multiple closing and opening operations to simplify but also aggregate the input polygons. To retain the original rectangular geometry that is typical for buildings they implement the closing operator with a Minkowski sum of the input polygon and a square aligned with it.

Our method for polygon aggregation computes a binary partition of the set of polygons of a planar subdivision. For computing the binary partition we adopt a technique from computer vision based on graph cuts [5]. Graph cuts are very commonly used for image segmentation [35, 39] or stereo matching [2]. However, most applications are concerned with raster data. An exception is the work by Sadlacek and Zara. [34] dealing with the generation of polygonal objects from three-dimensional point clouds. With respect to the automatic tuning of a parameter that balances two objectives, the work of Peng and Veksler [30] is most related to ours. While they focus on the development of quality metrics for the



(a) an instance of Problem 1, where  $B = \{p_1, p_6\}$ ; red (bold) edges delineate a solution with selection  $S' = \{p_1, p_3, p_4, p_6\}$ .



(b) graph  $G$  for the instance in (a); red (dashed) edges show an  $s$ - $t$ -cut modelling the solution  $S'$ .

■ **Figure 4** Algorithmic solution of Problem 1 via a graph cut.

evaluation of image segmentation solutions, they simply use a constant step width to sample different values for the parameter of a weighted-sum model. In contrast, we focus on a more systematic exploration of different parameter values.

### 3 Polygon Aggregation via Minimum Cuts

For solving Problem 1 with graph cuts, we set up an undirected weighted graph  $G = (V, E)$  modeling all feasible solutions as well as our minimization goal. This approach is illustrated in Fig. 4 and described in detail in the following.

As starting point we use the adjacency graph  $G' = (V', E')$  of the planar subdivision given with the set  $S$  of polygons. Assuming that the polygons are numbered in an arbitrary order as  $p_1, \dots, p_n$ , we refer to the corresponding nodes in  $V'$  as  $v_1, \dots, v_n$ . The edge set  $E'$  contains an edge  $\{v_i, v_j\}$  for every two polygons  $p_i$  and  $p_j$  whose boundaries share at least one line segment. We define the node set of  $G$  as  $V = V' \cup \{s, t\}$ , where  $s$  is a node called *source* and  $t$  a node called *sink*. The edge set  $E$  of  $G$  contains all edges in  $E'$  as well as, for  $i = 1, \dots, n$ , the two edges  $\{s, v_i\}$  and  $\{v_i, t\}$ ; see Figs. 4a and b. An  $s$ - $t$ -cut in  $G$  is a set of edges whose removal from  $G$  causes  $s$  and  $t$  to be in different connected components. We solve Problem 1 by defining an edge weighting  $w: E \rightarrow \mathbb{R}_{\geq 0}$  and computing a *minimum*  $s$ - $t$ -cut in  $G$ , i.e., an  $s$ - $t$ -cut in  $G$  of minimum total edge weight.

Formally, for any  $s$ - $t$ -cut  $C \subseteq E$ , we define its weight as  $w(C) = \sum_{e \in C} w(e)$  and the graph  $G_C = (V, E \setminus C)$ . We call the connected component of  $G_C$  containing  $s$  the *source component* and the connected component of  $G_C$  containing  $t$  the *sink component* of  $C$ . Moreover, we refer to the set of polygons represented by nodes in the source component as the *solution modeled* by  $C$ ; see the red edges in Figs. 4a and b. It remains to ensure that any solution modeled by a *minimum*  $s$ - $t$ -cut in  $G$  is feasible and optimal with respect to Problem 1. For this we define the edge weighting  $w$  as follows:

- For every edge  $e = \{v_i, v_j\}$ , we set  $w(e) = (1 - \alpha) \cdot \ell(p_i, p_j)$ , where  $\ell(p_i, p_j)$  is the length of the common boundary of polygons  $p_i$  and  $p_j$ .
- For every node  $v_i$  with  $p_i \notin B$ , we set  $w(\{s, v_i\}) = 0$  and  $w(\{v_i, t\}) = \alpha \cdot A(p_i) + (1 - \alpha) \cdot \ell(p_i)$ , where  $\ell(p_i)$  is the length of the boundary between  $p_i$  and the outer face (0 if  $p_i$  and the outer face are not adjacent).
- For every node  $v_i$  with  $p_i \in B$ , we set  $w(\{s, v_i\}) = \infty$  and  $w(\{v_i, t\}) = \alpha \cdot A(p_i) + (1 - \alpha) \cdot \ell(p_i)$ . This avoids that  $\{s, v_i\}$  is selected for the cut and thus ensures that  $v_i$  is in the source component. (In practice, we use a floating-point number format with a special value representing  $\infty$ .)

For computing a minimum  $s$ - $t$ -cut and the corresponding optimal solution to Problem 1 we then use a standard graph algorithm.

► **Theorem 1.** *The solution modeled by any minimum  $s$ - $t$ -cut in  $G$  is an optimal solution to Problem 1. This allows Problem 1 to be solved in  $O(n^2/\log n)$  time.*

**Proof.** We prove that (i) each selection  $S'$  with  $B \subseteq S' \subseteq S$  is modeled by an  $s$ - $t$ -cut in  $G$  whose total weight is  $\alpha \cdot A(S') + (1 - \alpha) \cdot P(S') = f_\alpha(S')$  and (ii) each  $s$ - $t$ -cut in  $G$  of total weight  $W \neq \infty$  models a solution whose objective value measured with  $f_\alpha$  is at most  $W$ . This together implies that any solution modeled by a minimum  $s$ - $t$ -cut in  $G$  is an optimal solution to Problem 1.

To show (i), let  $S'$  be an arbitrary solution with  $B \subseteq S' \subseteq S$  and  $C$  the cut defined as follows. For each  $p_i \in S'$ , we add edge  $\{v_i, t\}$  to  $C$ , which amounts to weight  $\alpha \cdot \sum_{p_i \in S'} A(p_i) + (1 - \alpha) \cdot \sum_{p_i \in S'} \ell(p_i)$ . Moreover, we add each edge  $\{v_i, v_j\} \in E'$  with  $p_i \in S'$  and  $p_j \notin S'$  to  $C$ , which amounts to weight  $(1 - \alpha) \cdot \sum_{\{v_i, v_j\} \in E''} \ell(p_i, p_j)$  where  $E'' = \{\{v_i, v_j\} \in E' \mid p_i \in S' \wedge p_j \notin S'\}$ . The set  $C$  is an  $s$ - $t$ -cut in  $G$  modeling  $S'$  because the nodes for polygons in  $S'$  plus node  $s$  constitute the source component of  $C$ . The weight of  $C$  is

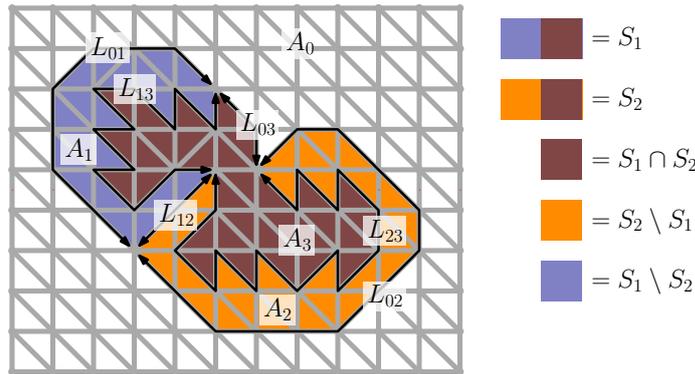
$$\begin{aligned} w(C) &= \alpha \cdot \sum_{p_i \in S'} A(p_i) + (1 - \alpha) \cdot \sum_{p_i \in S'} \ell(p_i) + (1 - \alpha) \cdot \sum_{\{v_i, v_j\} \in E''} \ell(p_i, p_j) \\ &= \alpha \cdot A(S') + (1 - \alpha) \cdot \left( \sum_{p_i \in S'} \ell(p_i) + \sum_{\{v_i, v_j\} \in E''} \ell(p_i, p_j) \right) \\ &= \alpha \cdot A(S') + (1 - \alpha) \cdot P(S') = f_\alpha(S'). \end{aligned}$$

To show (ii), we consider an arbitrary  $s$ - $t$ -cut  $C'$  in  $G$  of total weight  $w(C') = W \neq \infty$ . Let  $S'$  be the solution modeled by  $C'$ . Because of  $w(C') \neq \infty$ ,  $S'$  satisfies  $B \subseteq S' \subseteq S$ . Now, let  $C$  be the cut for  $S'$  as defined in the proof of (i). As argued before,  $C$  models  $S'$  and its weight equals the objective value of  $S'$ , i.e.,  $w(C) = \alpha \cdot A(S') + (1 - \alpha) \cdot P(S')$ . Moreover, the weight of  $C$  is at most the weight  $W$  of  $C'$ , since every edge in  $C$  is included in  $C'$  as well, which can be seen as follows. Assume that there exists an edge  $e = \{u, v\}$  with  $e \in C$  and  $e \notin C'$ . Because of  $e \in C$  and the way we constructed  $C$ , one of the nodes  $u$  and  $v$  lies in the source component of  $C$  and the other one in the sink component of  $C$ . Because of  $e \notin C'$ ,  $u$  and  $v$  lie in the same connected component of  $G_{C'}$ . This contradicts the assumption that  $C$  and  $C'$  model the same solution. Therefore, the assumption  $e \in C$  and  $e \notin C'$  must have been false. This allows us to conclude that  $w(C') \geq w(C) = \alpha \cdot A(S') + (1 - \alpha) \cdot P(S')$ .

The currently fastest algorithm for computing a minimum  $s$ - $t$ -cut in a graph with  $O(n)$  edges runs in  $O(n^2/\log n)$  time [29]. This result applies to our case since  $G'$  is planar (which implies that it has  $O(n)$  edges) and since  $G$  has only two more edges for each node of  $G'$ . The running time for computing the cut dominates the running times for computing  $G$  from the input and generating the output selection from the cut (e.g., via a depth-first search in the graph without the cut edges, starting from the source). Hence, the overall running time for solving Problem 1 is  $O(n^2/\log n)$ . ◀

#### 4 Computing Solutions for Multiple Parameter Values

In this section we deal with Problem 2, which asks for a set containing an  $\alpha$ -optimal solution for every  $\alpha \in [0, 1]$ . First, in Sect. 4.1, we show that a set with the required property and  $O(n)$  solutions exists. Then, in Sect. 4.2, we present an algorithm for computing such a set or, more generally, a set containing for every  $\alpha$  a solution that is at most a factor  $(1 + \varepsilon)$  worse than an  $\alpha$ -optimal solution while avoiding redundancy. The latter is relevant if the aim is to approximate the output set of Problem 2 with few solutions.



■ **Figure 5** Schematic visualization of two  $\alpha$ -optimal solutions  $S_1$  and  $S_2$ . The labels  $A_i$  refer to the areas and the labels  $L_{ij}$  refer to the line lengths used in the proof of Lemma 2.

### 4.1 Linear number of solutions

Our argument about a sufficient size for the set requested by Problem 2 is based on the following structural lemma.

► **Lemma 2.** *For an instance of Problem 1, let  $S_1$  be an  $\alpha_1$ -optimal solution and  $S_2$  be an  $\alpha_2$ -optimal solution, where  $\alpha_1 > \alpha_2$ . Then  $S_1 \subseteq S_2$ .*

**Proof.** We distinguish four classes of polygons: those that are neither in  $S_1$  nor in  $S_2$ ; those that are only in  $S_1$ ; those that are only in  $S_2$ ; and those that are in  $S_1 \cap S_2$ . We denote these classes by  $T_0 = S \setminus (S_1 \cup S_2)$ ;  $T_1 = S_1 \setminus S_2$ ;  $T_2 = S_2 \setminus S_1$ ; and  $T_3 = S_1 \cap S_2$ . By  $A_i$  we denote the area of  $T_i$ , and by  $L_{ij}$  we denote the total length of the edges that are shared by  $T_i$  and  $T_j$ ; see Fig. 5. Let  $\alpha'_i$  be  $1 - \alpha_i$ .

The  $\alpha_1$ -optimal solution  $S_1 = T_1 \cup T_3$  is at least as good as  $S_1 \cap S_2 = T_3$  for  $\alpha = \alpha_1$ :

$$\begin{aligned} \alpha_1(A_1 + A_3) + \alpha'_1(L_{01} + L_{03} + L_{12} + L_{23}) &\leq \alpha_1 A_3 + \alpha'_1(L_{03} + L_{13} + L_{23}) \\ \Leftrightarrow \alpha_1 A_1 + \alpha'_1 L_{12} &\leq \alpha'_1(L_{13} - L_{01}). \end{aligned} \tag{1}$$

The  $\alpha_2$ -optimal solution  $S_2 = T_2 \cup T_3$  is not worse than  $S_1 \cup S_2 = T_1 \cup T_2 \cup T_3$  for  $\alpha = \alpha_2$ :

$$\begin{aligned} \alpha_2(A_2 + A_3) + \alpha'_2(L_{02} + L_{03} + L_{12} + L_{13}) &\leq \alpha_2(A_1 + A_2 + A_3) + \alpha'_2(L_{01} + L_{02} + L_{03}) \\ \Leftrightarrow \alpha'_2(L_{13} - L_{01}) &\leq \alpha_2 A_1 - \alpha'_2 L_{12}. \end{aligned} \tag{2}$$

Now suppose  $S_1 \not\subseteq S_2$ , that is,  $A_1 > 0$ . With  $\alpha_1 > \alpha_2 \geq 0$ , Equation (1) then implies  $L_{13} - L_{01} > 0$ . Since  $\alpha_2 < \alpha_1$ , we also have  $\alpha'_1 < \alpha'_2$ . Combining this with (1) and (2) gives:

$$\alpha_1 A_1 \leq \alpha_1 A_1 + \alpha'_1 L_{12} \leq \alpha'_1(L_{13} - L_{01}) < \alpha'_2(L_{13} - L_{01}) \leq \alpha_2 A_1 - \alpha'_2 L_{12} \leq \alpha_2 A_1.$$

Thus,  $\alpha_1 A_1 < \alpha_2 A_1$ . However, this contradicts the assumption  $\alpha_2 < \alpha_1$ . Therefore, the assumption  $S_1 \not\subseteq S_2$  must have been false, and we conclude  $S_1 \subseteq S_2$ . ◀

Suppose that we continuously decrease  $\alpha$  from 1 to 0 while maintaining an  $\alpha$ -optimal solution in a lazy fashion, meaning that we replace the current solution only if it ceases being  $\alpha$ -optimal. At the beginning of this process we have the solution containing only the polygons in  $B$ , since for  $\alpha = 1$  area is all that matters. Due to Lemma 2, whenever we have to replace the current solution, we can define the new solution by including all the polygons selected in the current solution plus at least one additional polygon from the set  $S$ . Since there can be at most  $n - 1$  such events, we can conclude the following lemma.

► **Lemma 3.** *There exists a set of cardinality  $O(n)$  that contains an  $\alpha$ -optimal solution for every  $\alpha \in [0, 1]$ .*

## 4.2 The search algorithm

According to Lemma 3, a linear number of solutions suffices to have an  $\alpha$ -optimal solution for every  $\alpha \in [0, 1]$  as demanded by Problem 2. Our goal is now to find such a linear-size set efficiently within the set of all possible solutions, which has exponential size. We do this with a classical method for multi-objective optimization [8], which in the literature is sometimes referred to as *dichotomic scheme* [32] or *chord algorithm* [11]. In the following we describe the algorithm and how it is applied in our setting to approximate the result set of Problem 2. For this we assume a user-specified error tolerance  $\varepsilon \geq 0$  as input. We ask for an output set containing for every  $\alpha \in [0, 1]$  a solution  $S'$  satisfying  $f_\alpha(S') \leq (1 + \varepsilon) \cdot \text{opt}$ , where  $\text{opt}$  is the value of  $f_\alpha$  for an  $\alpha$ -optimal solution. Approximating for  $\varepsilon = 0$  will solve Problem 2.

Before describing the approximation algorithm we choose a geometric representation for solutions that simplifies the discussion. Consider visualizing a solution  $S'$  in a diagram with  $\alpha$  on the horizontal axis and the solution value  $\alpha A(S') + (1 - \alpha)P(S')$  on the vertical axis (see Fig. 6 for an illustration). Thus, the function graph of  $S'$  is a line through the points  $(0, P(S'))$  and  $(1, A(S'))$ . Now suppose we put the function graphs of all possible solutions in a single diagram. A solution  $S'$  is  $\alpha$ -optimal if the line for  $S'$  is the first line that is hit when we shoot a vertical ray from  $(\alpha, 0)$  upwards. Thus, to compute an  $\alpha$ -optimal solution for every  $\alpha$ , we need to compute the lower envelope (the piecewise minimum) of the arrangement of all possible solutions' lines, restricted to the  $\alpha$ -range  $[0, 1]$ ; see Fig. 6a. Moreover, approximating the set of all  $\alpha$ -optimal solutions roughly means to find a set of solutions whose lower envelope is not too far above the lower envelope of all solutions.

To solve the approximation problem, we use the recursive procedure APPROX presented in Algorithm 1, which only considers a range  $[\alpha_L, \alpha_U]$  of values for  $\alpha$ . To consider all values for  $\alpha$  in the range  $[0, 1]$ , we proceed as follows:

1. Compute  $\alpha$ -optimal solutions  $S_L$  for  $\alpha = 0$  and  $S_U$  for  $\alpha = 1$  using the min-cut algorithm.
2. Invoke APPROX with  $\alpha_L = 0$  and  $\alpha_U = 1$ : APPROX( $S, B, 0, 1, S_L, S_U, \varepsilon$ )

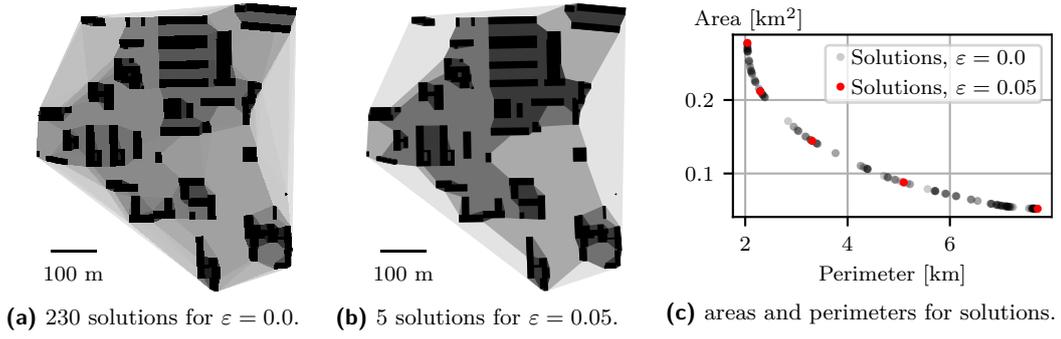
The solutions  $S_L$  and  $S_U$  are passed over to the procedure only to allow it to access their associated areas  $A(S_L)$  and  $A(S_U)$  as well as perimeters  $P(S_L)$  and  $P(S_U)$ . To keep the presentation simple, we refer to these as  $A_L$  and  $A_U$  as well as  $P_L$  and  $P_U$ , respectively.

The recursive procedure operates as follows. If  $A_L = A_U$  and  $P_L = P_U$  (this may happen in special cases), then the recursive procedure returns without output (ending in a leaf of the recursion tree). Otherwise, we compute the  $\alpha$ -value  $\alpha_c = (P_U - P_L)/(A_L - A_U + P_U - P_L)$  for which  $S_L$  and  $S_U$  are equally good. (In effect, we compute the  $\alpha$ -coordinate of the intersection point  $C$  of the lines representing  $S_L$  and  $S_U$ .) Then, we call the min-cut algorithm to obtain an  $\alpha_c$ -optimal solution  $S_C$ . We compare the objective function value for this new solution with the objective function for  $S_L$  (comparing to  $S_U$  would yield the same result). If the new solution is substantially better (depending on the approximation factor  $\varepsilon$ ), then the corresponding line passes far below the intersection point  $C$  in the diagram between  $\alpha_L$  and  $\alpha_U$  (refer to Fig. 6b for an illustration). We then call the procedure recursively twice: once with range  $(\alpha_L, \alpha_c)$  and the solutions  $S_L$  and  $S_C$ , and once with range  $(\alpha_c, \alpha_U)$  and the solutions  $S_C$  and  $S_U$ . In between these two calls, we output the new solution  $S_C$ . The described process is outlined by Algorithm 1.

For computing a complete set of solutions  $S_1, \dots, S_k$  representing the  $k$  linear pieces that form the lower envelope we call the algorithm with  $\varepsilon = 0$ . Correctness of the procedure can be shown by induction on the recursion tree. As for the running time, observe that, by



## 6:10 Bicriteria Aggregation of Polygons via Graph Cuts



■ **Figure 7** Solutions obtained with Algorithm 1 for the same instance without (a) and with approximation (b) and their values for the two objectives (c). Darker polygons correspond to solutions for higher  $\alpha$  values, which consist of multiple groups.

meaning that clusters but also larger gaps exist. Nevertheless, our approximation approach yields a small and representative set of solutions. Due to this result, we decided to focus on approximated result sets in the further experiments, which we present in the following.

We conducted further experiments with the aim to aggregate building polygons to settlement areas as defined in the digital landscape model DLM250 of the German “Authoritative Topographic-Cartographic Information System” (ATKIS). This corresponds to a map scale 1 : 250 000. We evaluate our approach on two instances of different sizes from the building data set, each of which roughly corresponds to one settlement polygon in the DLM250 but also includes buildings outside of the settlement; see Fig. 8. The larger instance consists of 16881 buildings and the smaller one of 853. They correspond to the town Euskirchen and the village Ahrem, respectively. Figure 9 depicts for Ahrem the influence of  $\varepsilon$  on the number of output solutions. Allowing a maximum error of  $\varepsilon = 0.01 = 1\%$  already decreases the number of solutions by several orders of magnitude in comparison to an exact solution ( $\varepsilon = 0.0$ ). For visualizing the aggregation results of the two instances we set  $\varepsilon = 0.1$ , resulting in six solutions for Euskirchen (see Figs. 8a and 10a) and five for Ahrem (see Figs. 8b and 10b).

To evaluate our aggregation results, we computed a rather accurate approximation for Euskirchen ( $\varepsilon = 10^{-6}$ ) and an exact solution to Problem 2 for Ahrem. We then compared every solution in the result set with the corresponding settlement polygon in the DLM250. For comparing a solution  $S_1$  with the reference  $S_2$  we used the following four different metrics:

- The *Jaccard index*, which is also known as *Intersection over Union (IoU)*, for polygons:

$$IoU = \frac{A(S_1 \cap S_2)}{A(S_1 \cup S_2)}$$

- The *Area Similarity* and *Perimeter Similarity*, as suggested by Podolskaya et al. [31] for quality assessment of polygon generalization:

$$V_A = 1 - \frac{|A(S_1) - A(S_2)|}{\max\{A(S_1), A(S_2)\}} \quad V_P = 1 - \frac{|P(S_1) - P(S_2)|}{\max\{P(S_1), P(S_2)\}}$$

- The *discrete Hausdorff distance*  $d_H$  of the polygons’ boundaries, whose vertex sets we denote as  $V(S_1)$  and  $V(S_2)$ . It adds to the other three measure that it indicates the maximum difference of the solutions with respect to the Euclidean distance  $d$ .

$$d_H = \max_{v_1 \in V(S_1)} \left\{ \min_{v_2 \in V(S_2)} \{d(v_1, v_2)\} \right\}$$

■ **Table 1** Statistics for every solution that is the **best** for at least one evaluation metric. For the Hausdorff distance lower is better ( $\downarrow$ ), else higher is better ( $\uparrow$ ).

Dataset	$\alpha$	$IoU \uparrow$	$V_A \uparrow$	$V_P \uparrow$	$d_H [m] \downarrow$
Euskirchen	0.0048	0.8951	<b>0.9997</b>	0.8033	298.476
	0.0064	<b>0.8999</b>	0.9809	0.8437	273.568
	0.0073	0.8943	0.9650	0.8495	<b>264.562</b>
	0.0109	0.8301	0.8795	<b>0.9989</b>	926.226
Ahrem	0.0021	0.5882	0.6973	<b>0.9998</b>	336.437
	0.0041	<b>0.7896</b>	<b>0.9685</b>	0.9106	<b>114.570</b>

The similarity measures  $IoU$ ,  $V_A$ , and  $V_B$  are in the range of  $[0, 1]$ , where 0 represents minimum similarity and 1 represents maximum similarity. For cases where one polygon is contained by a second polygon,  $V_A$  equals  $IoU$ .

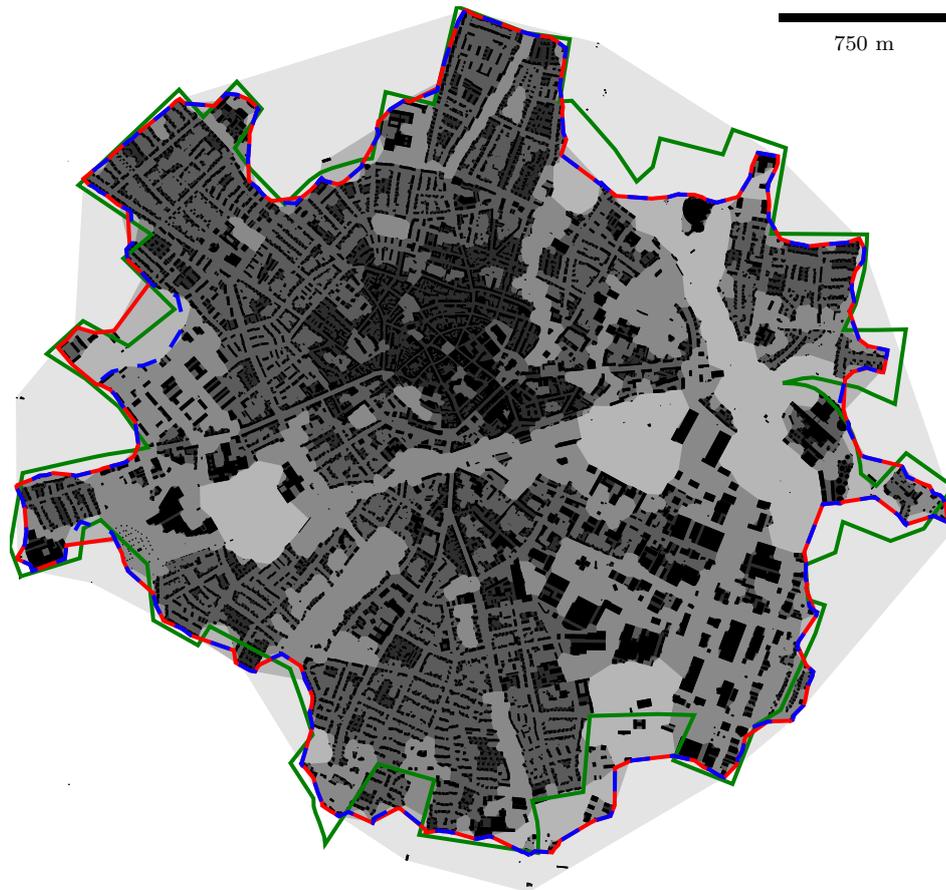
An important selection criterion of the DLM250 is that all settlement areas must be larger than 40 hectares. Therefore, we removed all polygons smaller than the threshold from a solution before computing the four metrics. The resulting metrics for Euskirchen are shown in Fig. 11. For all  $\alpha > 0.105$  the solution contained no contiguous polygon reaching the 40 hectares threshold, meaning that no similarity can be found. The graph reveals a correlation of  $IoU$  and Area Similarity  $V_A$ ; in fact they are the same for  $\alpha > 0.05$ . The Perimeter Similarity  $V_P$ , on the other hand, turned out to be quite meaningless as its graph shows a rather erratic behaviour. This is because solutions for large  $\alpha$  often contain polygons smaller than the threshold, leading to a short perimeter after their removal.

For each metric we identified the value for  $\alpha$  that yields the solution of maximum similarity or minimum Hausdorff distance; see Table 1. For the cases where  $IoU$ ,  $V_A$ , and Hausdorff distance have their respective best values, the other metrics are similar to their best values. At the maximum of  $V_P$ , the other metrics are significantly away from their best values. The solutions of minimum Hausdorff distance and maximum  $IoU$  are also depicted in Figure 8 as blue and red lines, respectively. We observe that the shapes are indeed similar, but that the boundary of the reference solutions contain more angles close to  $90^\circ$ . Using a schematization algorithm in a post-processing step might increase the similarity in this respect.

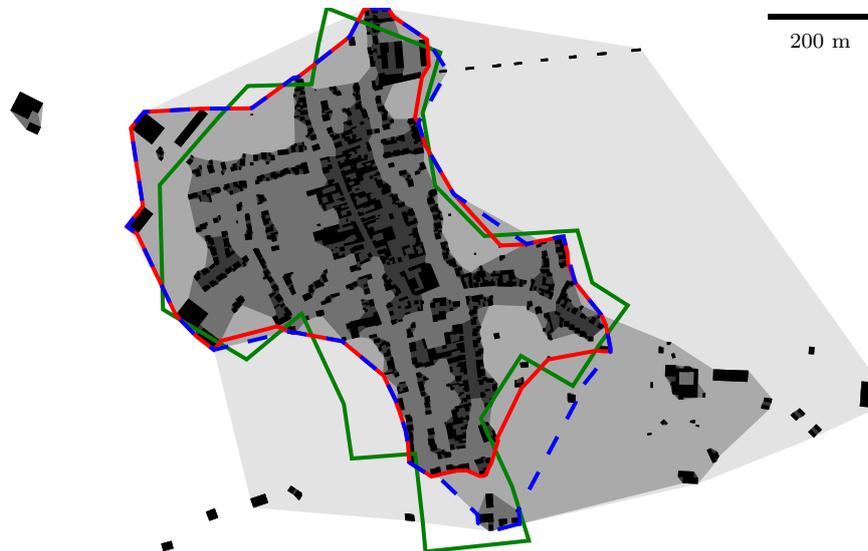
In our Java implementation, we used the Push Relabel algorithm implemented in the library JGraphT [25] to compute the graph cuts. This algorithm can be implemented to run in the worst case in  $O(nm \log(n^2/m))$  time on a graph with  $n$  nodes and  $m$  edges [17], but the JGraphT documentation reports a worst-case running time of  $O(n^3)$ . This implies that our implementation of the algorithm for Problem 1 runs in  $O(n^3)$  time, where  $n$  is the number of polygons in the set  $S$ . This is substantially higher than the sub-quadratic running time that is achievable according to Theorem 1, but sufficiently low to solve problem instances as the ones we discussed above. Our implementation of the recursive algorithm for Problem 2 uses parallelization for the two recursive calls, which turned out to improve the running time substantially. In particular, the running time improves from 79 seconds to 12 seconds for  $\varepsilon = 10^{-6}$  (528 solutions) on the instance of Ahrem by using parallelization.

## 6 Conclusion and Outlook

We have presented efficient algorithms for polygon aggregation optimizing a balance between a small total area and a short total perimeter of the output polygons. We combined the two criteria in a weighted sum, which we parameterized with a single parameter  $\alpha \in [0, 1]$ . The



(a) Euskirchen.



(b) Ahrem.

■ **Figure 8** Result sets of two evaluation data sets with  $\varepsilon = 0.1$ . The green outline corresponds to the ground truth polygon. The red and blue outlines, which are mostly the same, represent the best solutions in terms of  $IoU$  and  $d_H$ .

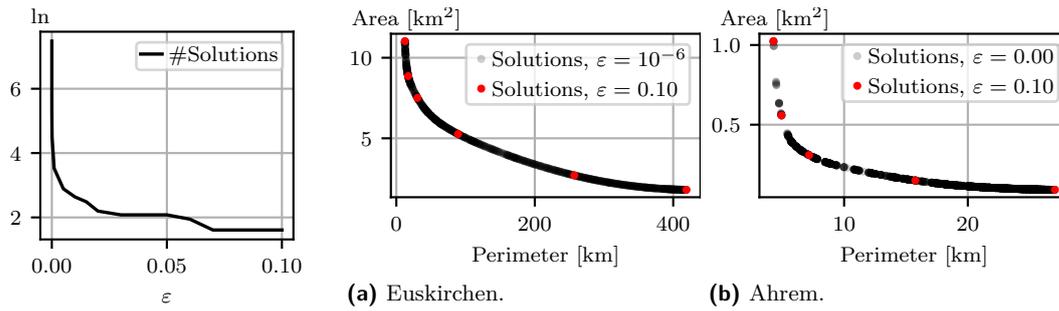


Figure 9 Number of solutions by  $\varepsilon$  for Ahrem.

Figure 10 All  $\alpha$ -optimal solutions and the approximated solutions which are displayed in Fig. 8.

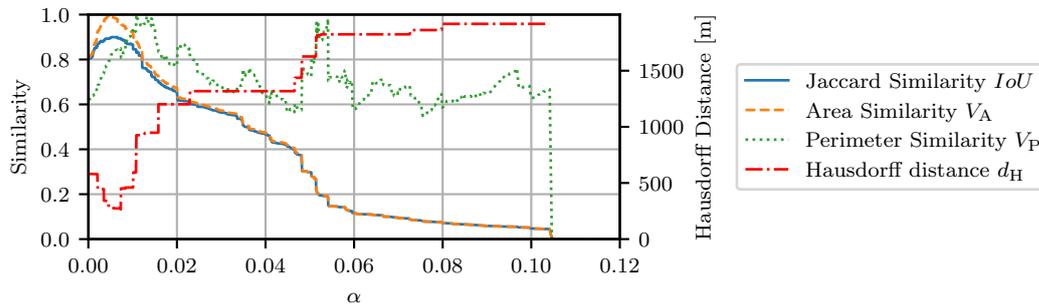


Figure 11 Evaluation metrics for Euskirchen shown in Fig. 8a. For  $\alpha > 0.105$  all polygon of the result set are below the specified minimum area.

first problem we studied asked for a single optimal solution for a fixed  $\alpha$ . It turned out that this problem can be solved by computing a minimum cut in a graph. A second problem asked for an output set containing an optimal solution for every possible value for  $\alpha$ . We showed that a linear-size set with the requested property can be efficiently computed with a recursive algorithm that uses the graph-cut algorithm as a subroutine. Moreover, we showed how to approximate such a set using the same recursive algorithm.

Our experiments showed that the presented algorithms are fast enough to process realistic problem instances, although we did not use the fastest known (i.e., sub-quadratic) graph-cut algorithm in our implementation. We consider it astonishing how few solutions were needed to approximate a set containing an optimal solution for every  $\alpha$ : For our largest instance with 16881 building footprints, a set of six solutions sufficed to include for every  $\alpha$  a solution that is at most 10% worse than optimal. Since the number of graph cuts computed by the recursive algorithm is in the order of the size of its output set, the approximation for the above-mentioned instance was achieved relatively fast, in 20.3 seconds. Finally, our experiments support the claim that our algorithm can aggregate building footprints to polygons that are quite similar to settlement areas as given in an official topographic database of scale 1 : 250 000. However, experiments on a larger data basis are needed to substantiate our finding. It would also be interesting to investigate further whether different instances can be solved with the same  $\alpha$  to obtain a solution similar to a reference solution.

As an idea for future research one could replace the triangulation used in our method with other partitions of the plane. For example, one could try to preserve regularities of the input polygons by subdividing the plane with linear extensions of the polygons' edges. Moreover, it would be interesting to consider relaxed or more constrained versions of the aggregation

problem. For example, one could relax the requirement to include every input polygon in the output and/or introduce a hard size constraint for the output polygons. Most importantly, however, we see our work as a step towards multi-criteria optimization in cartography. As next steps one could consider more than two criteria or look at Pareto-optimal solutions rather than just at the extreme nondominated (i.e.,  $\alpha$ -optimal) solutions.

---

## References

- 1 Karl-Heinrich Anders and Monika Sester. Parameter-free cluster detection in spatial databases and its application to typification. In *Proc. 19th ISPRS Congress*, volume 33 of *International Archives of Photogrammetry and Remote Sensing*, pages 75–83, 2000.
- 2 Michael Bleyer and Margrit Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing: Image Communication*, 22(2):127–143, 2007. doi:10.1016/j.image.2006.11.012.
- 3 Fritz Böckler and Petra Mutzel. Output-sensitive algorithms for enumerating the extreme nondominated points of multiobjective combinatorial optimization problems. In *Proc. 23rd Annual European Symposium on Algorithms (ESA '15)*, pages 288–299, 2015. doi:10.1007/978-3-662-48350-3\_25.
- 4 Annika Bonerath, Benjamin Niedermann, and Jan-Henrik Haurert. Retrieving  $\alpha$ -shapes and schematic polygonal approximations for sets of points within queried temporal ranges. In *Proc. 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '19)*, pages 249–258, 2019. doi:10.1145/3347146.3359087.
- 5 Yuri Boykov and Olga Veksler. Graph cuts in vision and graphics: Theories and applications. In *Handbook of Mathematical Models in Computer Vision*, chapter 5, pages 79–96. Springer, Boston, MA, USA, 2006. doi:10.1007/0-387-28831-7\_5.
- 6 Dirk Burghardt and Alessandro Cecconi. Mesh simplification for building typification. *International Journal of Geographical Information Science*, 21(3):283–298, 2007. doi:10.1080/13658810600912323.
- 7 Dirk Burghardt, Stefan Schmid, and Jantien Stoter. Investigations on cartographic constraint formalisation. In *Proc. 11th ICA Workshop on Generalisation and Multiple Representation*, 2007. URL: <https://kartographie.geo.tu-dresden.de/downloads/ica-gen/workshop2007/Burghardt-ICAWorkshop.pdf>.
- 8 Jared L. Cohon. *Multiobjective Programming and Planning*. Academic Press, New York, NY, USA, 1978.
- 9 Jonathan Damen, Marc van Kreveld, and Bert Spaan. High quality building generalization by extending the morphological operators. In *Proc. 11th ICA Workshop on Generalisation and Multiple Representation*, pages 1–12, 2008. URL: [https://kartographie.geo.tu-dresden.de/downloads/ica-gen/workshop2008/04\\_Damen\\_et\\_al.pdf](https://kartographie.geo.tu-dresden.de/downloads/ica-gen/workshop2008/04_Damen_et_al.pdf).
- 10 Steven J. D’Amico, Shouu-Jiun Wang, Rajan Batta, and Christopher M. Rump. A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6):667–684, 2002. doi:10.1016/S0305-0548(01)00056-9.
- 11 Constantinos Daskalakis, Ilias Diakonikolas, and Mihalis Yannakakis. How good is the chord algorithm? *SIAM Journal on Computing*, 45(3):811–858, 2016. doi:10.1137/13093875X.
- 12 Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236, 2008. doi:10.1016/j.patcog.2008.03.023.
- 13 Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983. doi:10.1109/TIT.1983.1056714.
- 14 Yu Feng, Frank Thiemann, and Monika Sester. Learning cartographic building generalization with deep convolutional neural networks. *ISPRS International Journal of Geo-Information*, 8(6), 2019. doi:10.3390/ijgi8060258.

- 15 Martin Galanda. *Automated polygon generalization in a multi agent system*. PhD thesis, University of Zurich, Zürich, 2003. doi:10.5167/uzh-163108.
- 16 Sven Gedicke, Johannes Oehrlein, and Jan-Henrik Haunert. Aggregating land-use polygons considering line features as separating map elements. *Cartography and Geographic Information Science*, 48(2):124–139, 2021. doi:10.1080/15230406.2020.1851613.
- 17 Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988. doi:10.1145/48014.61051.
- 18 Jan-Henrik Haunert and Alexander Wolff. Area aggregation in map generalisation by mixed-integer programming. *International Journal of Geographical Information Science*, 24(12):1871–1897, 2010. doi:10.1080/13658810903401008.
- 19 Christopher B. Jones, Geraint Ll. Bundy, and Mark J. Ware. Map generalization with a triangulated data structure. *Cartography and Geographic Information Systems*, 22(4):317–331, 1995. doi:10.1559/152304095782540221.
- 20 Kamyoun Kim, Denis J. Dean, Hyun Kim, and Yongwan Chun. Spatial optimization for regionalization problems with spatial interaction: a heuristic approach. *International Journal of Geographical Information Science*, 30(3):451–473, 2016. doi:10.1080/13658816.2015.1031671.
- 21 Chengming Li, Yong Yin, Xiaoli Liu, and Pengda Wu. An automated processing method for agglomeration areas. *ISPRS International Journal of Geo-Information*, 7(6):204, 2018. doi:10.3390/ijgi7060204.
- 22 Jingzhong Li and Tinghua Ai. A triangulated spatial model for detection of spatial characteristics of GIS data. In *Proc. 2010 IEEE International Conference on Progress in Informatics and Computing (PIC '10)*, volume 1, pages 155–159, 2010. doi:10.1109/PIC.2010.5687417.
- 23 Adrien Maudet, Guillaume Touya, Cécile Duchêne, and Sébastien Picault. Multi-agent multi-level cartographic generalisation in CartAGen. In *Proc. 12th International Conference on Advances in Practical Applications of Heterogeneous Multi-Agent Systems (PAAMS '14)*, pages 355–358, 2014. doi:10.1007/978-3-319-07551-8\_37.
- 24 Robert B. McMaster and K. Stuart Shea. *Generalization in digital cartography*. Association of American Geographers, Washington, DC, USA, 1992.
- 25 Dimitrios Michail, Joris Kinable, Barak Naveh, and John V. Sichi. JGraphT—a Java library for graph data structures and algorithms. *ACM Transactions on Mathematical Software*, 46(2), 2020. doi:10.1145/3381449.
- 26 Adriano Moreira and Maribel Y. Santos. Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. In *Proc. 2nd International Conference on Computer Graphics Theory and Applications (GRAPP '07)*, pages 61–68, 2007. URL: <http://hdl.handle.net/1822/6429>.
- 27 Johannes Oehrlein and Jan-Henrik Haunert. A cutting-plane method for contiguity-constrained spatial aggregation. *Journal of Spatial Information Science*, 15(1):89–120, 2017. doi:10.5311/JOSIS.2017.15.379.
- 28 OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>, 2020.
- 29 James B. Orlin. Max flows in  $O(nm)$  time, or better. In *Proc. 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, page 765–774, 2013. doi:10.1145/2488608.2488705.
- 30 Bo Peng and Olga Veksler. Parameter selection for graph cut based image segmentation. In *Proc. British Machine Vision Conference (BMVC '08)*, pages 16.1–16.10, 2008. doi:10.5244/C.22.16.
- 31 Ekaterina S. Podolskaya, Karl-Heinrich Anders, Jan-Henrik Haunert, and Monika Sester. Quality assessment for polygon generalization. In *Quality Aspects in Spatial Data Mining*, chapter 16, pages 211–220. CRC Press, Boca Raton, FL, USA, 2007. doi:10.1201/9781420069273.ch16.
- 32 Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. *INFORMS Journal on Computing*, 22(3):371–386, 2010. doi:10.1287/ijoc.1090.0342.

## 6:16 Bicriteria Aggregation of Polygons via Graph Cuts

- 33 Azimjon Sayidov, Robert Weibel, and Stefan Leyk. Recognition of group patterns in geological maps by building similarity networks. *Geocarto International*, pages 1–20, 2020. doi:10.1080/10106049.2020.1730449.
- 34 David Sedlacek and Jiri Zara. Graph cut based point-cloud segmentation for polygonal reconstruction. In *Proc. 5th International Symposium on Advances in Visual Computing (ISVC '09)*, pages 218–227, 2009. doi:10.1007/978-3-642-10520-3\_20.
- 35 Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi:10.1109/34.868688.
- 36 Stefan Steiniger, Dirk Burghardt, and Robert Weibel. Recognition of island structures for map generalization. In *Proc. 14th Annual ACM International Symposium on Advances in Geographic Information Systems (ACM GIS '06)*, pages 67–74, 2006. doi:10.1145/1183471.1183484.
- 37 Guillaume Touya, Xiang Zhang, and Imran Lokhat. Is deep learning the new agent for map generalization? *International Journal of Cartography*, 5(2–3):142–157, 2019. doi:10.1080/23729333.2019.1613071.
- 38 Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1992. doi:10.1119/1.14057.
- 39 Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993. doi:10.1109/34.244673.

# Coordinated Schematization for Visualizing Mobility Patterns on Networks

**Bram Custers** ✉ 

Eindhoven University of Technology, The Netherlands

**Wouter Meulemans** ✉ 

Eindhoven University of Technology, The Netherlands

**Bettina Speckmann** ✉ 

Eindhoven University of Technology, The Netherlands

**Kevin Verbeek** ✉ 

Eindhoven University of Technology, The Netherlands

---

## Abstract

GPS trajectories of vehicles moving on a road network are a valuable source of traffic information. However, the sheer volume of available data makes it challenging to identify and visualize salient patterns. Meaningful visual summaries of trajectory collections require that both the trajectories and the underlying network are aggregated and simplified in a coherent manner. In this paper we propose a coordinated fully-automated pipeline for computing a schematic overview of mobility patterns from a collection of trajectories on a street network. Our pipeline utilizes well-known building blocks from GIS, automated cartography, and trajectory analysis: map matching, road selection, schematization, movement patterns, and metro-map style rendering. We showcase the results of our pipeline on two real-world trajectory collections around The Hague and Beijing.

**2012 ACM Subject Classification** Theory of computation → Computational geometry

**Keywords and phrases** Trajectories, Visualization, Schematization

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.7

**Supplementary Material** *Software (Source code)*: <https://github.com/tue-alga/CoordinatedSchematization>; archived at [swh:1:dir:a04dfee52bf07c76d37ac692bc17450feec728df](https://www.swh.io/dir/a04dfee52bf07c76d37ac692bc17450feec728df)

**Funding** *Bram Custers*: Supported by HERE Technologies and the Dutch Research Council (NWO); 628.011.005.

**Acknowledgements** We would like to thank HERE Technologies for providing the HR dataset.

## 1 Introduction

GPS tracks from moving vehicles are an important source of information for traffic analysis and urban planning. Their general ubiquity allows decision makers to understand the usage of transportation networks and urban spaces. However, the sheer volume of the data makes it challenging to render trajectory collections in a meaningful way as to show the general, overarching patterns. Simply plotting all trajectories results in the infamous “spaghetti heaps”. Heat maps [15, 19] and other aggregation techniques such as Voronoi aggregation [1] are helpful to “untangle” traffic locally, but they generally fail to capture structural patterns, such as important longer routes.

Summarizing trajectory collections visually, such that salient patterns emerge, inherently requires a form of aggregation or simplification of the data. That is, the level of detail and information shown should be scale-appropriate and avoid a cognitive overload, while still being able to provide insight into the overall mobility. There are various techniques to cluster trajectories and compute a representative for visualization [6, 16], or to simplify



© Bram Custers, Wouter Meulemans, Bettina Speckmann, and Kevin Verbeek; licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Versteegen; Article No. 7; pp. 7:1–7:16

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

trajectories [12]. However, these techniques typically focus on trajectories in a general 2D space, whereas our focus lies on trajectory data on transportation networks, specifically vehicles on a road network. As such, the problem changes in nature, as selecting representative routes and performing simplification needs to be “network-aware”. Stronger still, to reduce the visual complexity of the eventual visualization, not only the trajectories need to be simplified, but also the underlying street network. To arrive at meaningful results which show traffic patterns in the correct context, the simplification and aggregation of the trajectory collection and of the network have to go hand-in-hand: they need to be *coordinated*.

**Contribution and organization.** We propose a coordinated fully-automated pipeline for computing a schematic overview of mobility patterns. Our pipeline consists of five steps, each utilizing well-known building blocks from GIS, automated cartography and trajectory analysis: map matching, road selection, schematization, movement patterns, and metro-map style rendering. We present our overall pipeline and its rationale in Section 2; the subsequent sections describe each step in more detail. Each of these sections also describes how we implemented the corresponding step in our proof-of-concept. For illustration we use a real-world dataset of vehicle trajectories around The Hague in the Netherlands. In Section 8 we discuss the results of our pipeline using a second real-world data set around Beijing. We close with a general discussion of our pipeline and future work in Section 9.

**Related work.** We focus here on related work pertaining to the visualization of large volumes of trajectories and discuss related work for each step of the pipeline in the respective section. Most research in this area aims to provide an overview of space usage, without showing or using the temporal component of trajectories; see the two extensive surveys by Chen et al. [10] and Andrienko et al. [3]. The notable exception are space-time cubes [13, 22], though they do not scale well to large numbers of trajectories without some form of aggregation.

To identify larger patterns, one can focus on visualizing the origin-destination data only, that is, focus only on the endpoints of the trajectories, possibly with some form of spatial aggregation. There are various techniques to visualize such information, e.g., [24, 31, 34]. Visualizing OD-data shows patterns beyond local traffic, but typically does not show any information on the actual routes. As such it does not support understanding mobility from the viewpoint of traveling through a network. Indeed, these techniques are typically applied in situations where the exact trajectories or routes are not available or not of interest.

## 2 The pipeline

Our input is a set  $\mathcal{T}$  of trajectories, and a network  $G$ . Our goal is a schematized representation of  $G$  together with the most salient mobility patterns in  $\mathcal{T}$ . Before we can describe our pipeline in more detail, we first give the necessary definitions.

**Definitions.** A *trajectory* is a sequence of measurements  $T = \langle (x_1, y_1, t_1), \dots, (x_n, y_n, t_n) \rangle$  with a position  $(x_i, y_i) \in \mathbb{R}^2$  at each timestamp  $t_i \in \mathbb{R}$ . We restrict our attention to the spatial domain and hence we ignore the temporal component beyond providing an ordering of the measurements. Thus, for the purpose of this paper, a trajectory  $T$  is a (directed) polygonal line with vertices  $(x_i, y_i)$ .

A (*road*) *network* is a directed graph  $G = (V, E)$  where each vertex has a location in  $\mathbb{R}^2$ . Edges have an associated geometry connected to their endpoints; initially, this is typically the unique line segment, but during our pipeline, this geometry can change. We further assume that each road has an associated road type, which is ordinal (e.g., “highway”).

We assume that the network is plane, that is, the edges do not cross except at common endpoints. This condition is not generally satisfied by actual road networks, but we can introduce extra vertices on these intersections to planarize the network. By marking these extra vertices, we can perform algorithms both on the planarized and original network. We further use  $|G|$  to denote the complexity of the network, measured as its number of edges, though note that this assumption implies that  $|G| = |E| = O(|V|)$ .

We use *route* to refer to a (directed) path in the network. For example, map-matching a trajectory results in its route: the path in the network that was traversed by the vehicle captured by the (noisy) trajectory. We refer to such a path as *the* route of the trajectory.

A route does not have to correspond to a single or entire trajectory. Specifically, we say that a route is *supported by* a trajectory  $T$  if it is a subpath of the route of  $T$ . We use *bundle* to indicate a route that is supported by multiple trajectories, using the *support of a bundle* to indicate the number of supporting trajectories. Bundles should aim to capture mobility patterns; precise criteria to form meaningful bundles are discussed in Section 6.

**The pipeline.** Our goal is to compute a schematic representation of  $G$  with salient bundles that are “supported by”  $\mathcal{T}$ . To achieve this, our pipeline consists of five steps, briefly sketched below. See Fig. 1 for an example of the results of these steps. In the subsequent sections we discuss each in more detail. Important in our treatment of the network and the trajectory information is to coordinate changes: specifically, changes in the network should be translated to changes in the trajectory information.

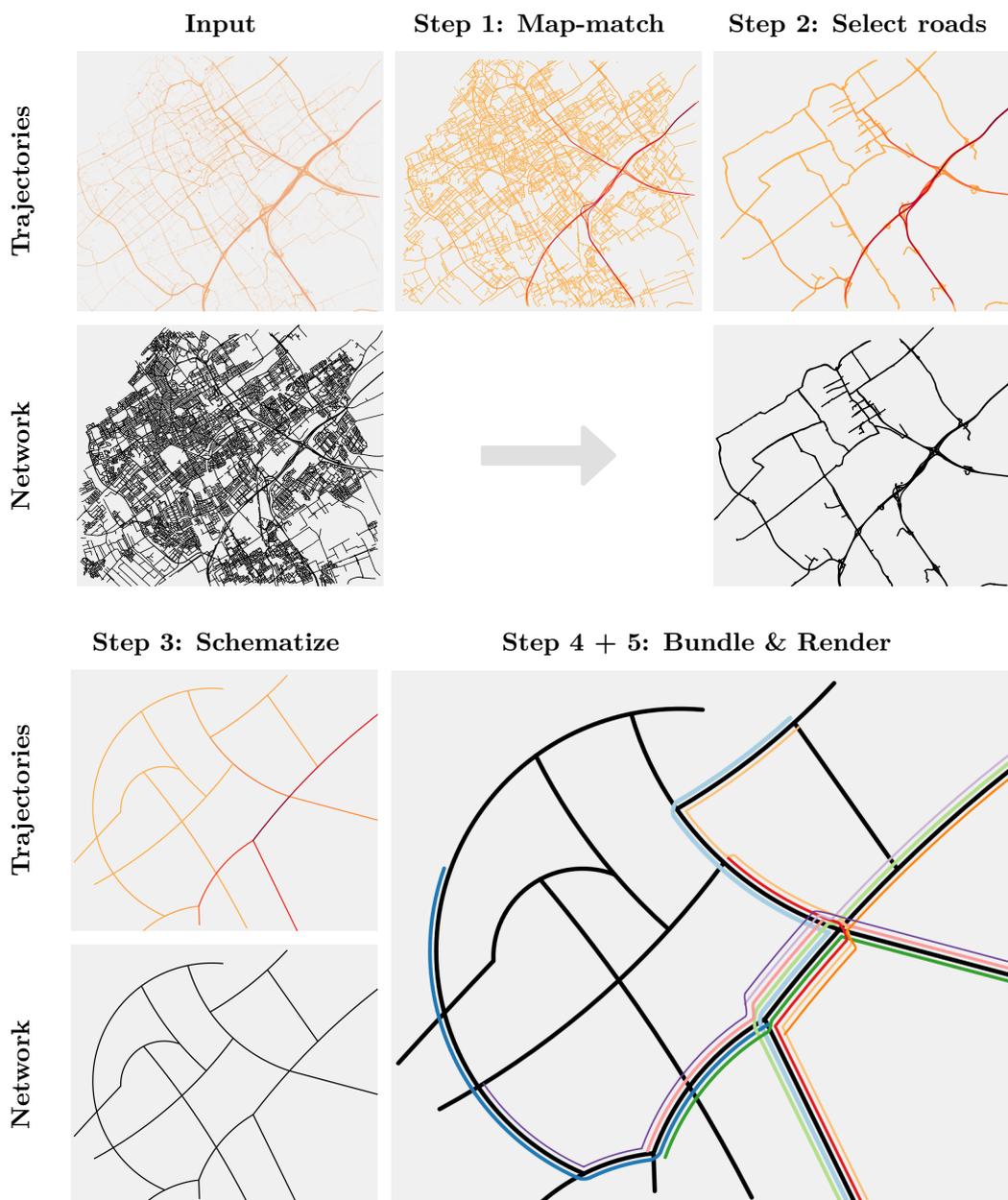
**Step 1: Map-match** We aim to visualize mobility patterns via bundles, frequent routes in the data. However, trajectories are not the same as routes, and thus cannot support a bundle. As such, we first map-match the trajectories to the network. The minimal input to this step is a single trajectory and the network, such that each trajectory can be processed individually. Map-matching computes the route associated with this trajectory. The result of this step is a set  $\mathcal{R}$  of routes, rather than trajectories. In our implementation, the trajectories are not used further. This step enables our pipeline to coordinate changes in the network and the routes.

**Step 2: Select roads** A typical road network is very detailed, much beyond the level of detail that we need to visualize the general mobility patterns and well-supported bundles. The minimal input to this step is the road network and the set of routes derived in Step 1. Note that we could, in principle, base selection purely on the network, and adapt the routes as necessary. However, we also choose to include all parts of the network which are frequented heavily by the routes. The result is a subset of the network and a mapping of the original routes to this selected network.

**Step 3: Schematize** To reinforce the summarizing nature of the eventual visualization, we reduce the visual complexity of the selected network via schematization. The input is the selected network and the mapping of the routes. The output is a strongly simplified version of this network. The mapping of the routes is maintained (coordinated) during this process. Optionally, the edges of this schematic network may be annotated with information about the length of the edge for the purpose of bundling.

**Step 4: Bundle** In our schematic representation, we find well-supported bundles. The input is thus the schematic network and the mapping of routes to this schematic representation. The output is a set of bundles that are well supported.

**Step 5: Render** We now have all ingredients for our visualization: the schematic network as well as salient mobility patterns (bundles). The result is the eventual visualization which shows these two pieces of information effectively.



■ **Figure 1** Our pipeline for coordinated schematization on the The Hague dataset. The input trajectories are shown as a density map. For the map-matched routes, we use an orange to red scale to convey low to high traffic volume per edge. We compute the bundles in Step 4 with  $S_{min} = 500$ ,  $L_{min} = 10000m$ ,  $p = 0.5$  and shrunk edge lengths, see Section 6 for more details on these parameters.

### 3 Step 1: Map-match trajectories to the network

**Desiderata.** To eventually visualize common routes in the network, we must ensure that our trajectory information is mapped to the network, that is, that each trajectory is translated into a route. This problem is generally known as *map matching*.

**Related work.** Map-matching is a broadly studied topic in GIS; see [8] for a recent survey. As discussed there, the different approaches are categorized by their matching model. For our purpose, we consider a map-matching approach using the similarity model, since this requires less parameters. But in principle, any approach will work in our pipeline.

**Our implementation.** We use the map-matching algorithm described by Alt et al. [2], which runs in  $O(|G||T| \log |G||T| \log |G|)$  time on a network  $G$  and trajectory  $T$ . The algorithm ensures that the route found has minimal Fréchet distance to the trajectory and thus that it is geometrically close. Our motivation for doing so is to remain as close as possible to the information of the trajectory. That is, to the information “visible” if we were to simply draw all trajectories. This algorithm is able to handle noise relatively well, but in case of very sparsely sampled trajectory data, it may struggle to find the most natural route.

#### 4 Step 2: Select roads

**Desiderata.** We aim to select the roads for two somewhat distinct purposes. First, we want to select the roads where there is considerable traffic, to facilitate well-supported bundles. This purpose is thus inherently data-driven. But second, we want to select major roads to provide a frame of reference for the viewer as to how the mobility patterns are situated in space. It stands to reason that often, major roads also carry a large part of the traffic. However, this is not necessarily the case.

**Related work.** Selection is an important part of road network generalization algorithms. The goal is to select the most important parts of the network, such that the remainder can be discarded in the simplification process. Different approaches exist to determine what features of the road-network are “salient”. Examples of these approaches are using the mesh density [9], using user defined weights [14] and using areas of faces combined with semantic labels [20]. Different from the previous are approaches that focus on “strokes” through the network: lines of good continuation, that is, lines with small local curvature [21]. During generalization, these strokes are considered atomic units and are selected based on their relative importance. This importance is often determined via network centrality measures [29, 33].

More recently, approaches focus on using traffic data to inform the selection process [23, 35]. Yu et al. propose an approach that is based on strokes, but during the selection process considers traffic flow from one stroke to the other, increasing the likelihood that strokes that give good traffic flow continuation are selected together. Van de Kerkhof et al. [23] follow a different approach, where the selection process is formulated as a covering problem, and the trajectories need to be covered by the selection of the road-network.



■ **Figure 2** Selection by road type (left) and by traffic (middle), combination of both (right).

**Our implementation.** We use the approach by van de Kerkhof et al. [23] which seeks to select a subgraph  $G'$  of the network with bounded length, such that the number of routes that are completely within this subgraph is maximized. Though the authors prove that this problem is NP-hard, they also describe a heuristic that runs in  $O(|\mathcal{T}|^2 \log |\mathcal{T}| + |\mathcal{T}||G'|)$  time; we use this heuristic in our implementation (see Fig. 2 (middle)). After selecting  $G'$ , we add any edges that were not selected yet and have a large enough road type. If the resulting selected network is not connected, we optionally select the largest connected component; none of the later steps require the network to be connected (see Fig. 2 (left) and (right)).

## 5 Step 3: Schematize the network

**Desiderata.** Even after selection, the network tends to contain more detail than necessary to provide a meaningful overview of mobility patterns: different lanes, cloverleaves, etc. Instead, we should get a high-level overview that communicates the main connectivity in the network, and as such create space to visualize mobility patterns (bundles). That is, we should collapse (aggregate) and simplify such local details. We do so beyond the need of target scale, instead focusing on *functional* detail: that is, we schematize the network. The network should remain spatially informative: roughly similar to the overall input geometry.

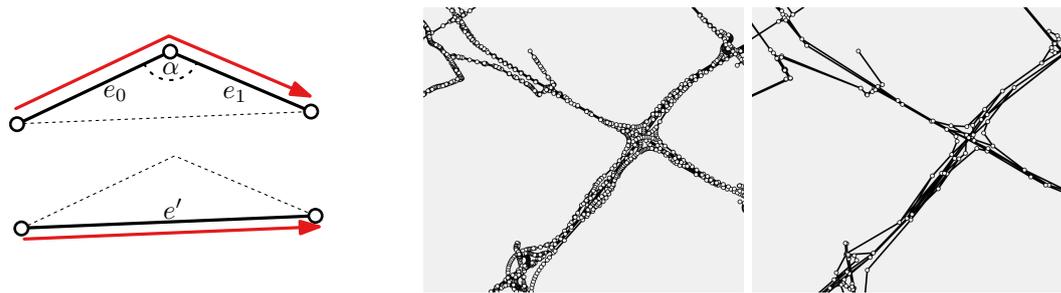
**Related work.** In automated cartography, the process of schematization is used to render aesthetically pleasing networks or polygonal domains that are decluttered enough to convey important information on the schematic [17]. Compared to generalization, schematization commonly reduces the input to such an extent that it is not necessarily realistic anymore, albeit retaining important features to recognize the original. Note that in general, the input to these algorithms is a detailed map of the road-network, whereas we input a selection of the map based on data, thus making it a data-driven schematization.

One approach is to limit the type of the geometry in the output, thus naturally reducing detail. Here it is common to fix the allowed number of orientations of lines in the network [7, 30], particularly in the context of metro maps. Alternatively one can fix the geometric primitives that can be used, for instance circular arcs [25] or Bézier curves [26].

To retain recognizability, schematization typically limits the spatial distortion between input and output by fixing vertex locations [7] or minimizing the distance between input and output edges, for instance via the Fréchet distance [25]. In addition, it is common to maintain the topology of the input, which plays a key role in recognition of the output. We note that for our approach, we want to retain the topology of the network at a certain scale, thus small topological features should be removed prior to applying a topology-preserving schematization approach. An alternative would be to consider continuous scale generalizations [27, 28] and applying schematization at the desired scale. An important aspect is then to be able to retain a mapping from the edges in the selected network to the schematization.

**Our implementation.** We first drastically simplify and collapse the selected network  $G'$ , after which we apply the arc schematization algorithm by van Dijk et al. [25] for its aesthetic and clean representation, resulting in a schematic road-network  $\mathcal{G}$ .

Our implementation applies the sequence of operations described below. We use simple steps in an incremental fashion to facilitate coordination and maintain a mapping between the edges of the selected network and the schematic network. Our simple operations can result in fairly coarse approximations. However, since our target is a highly abstracted final map, the coarseness of the earlier operations is not an issue.



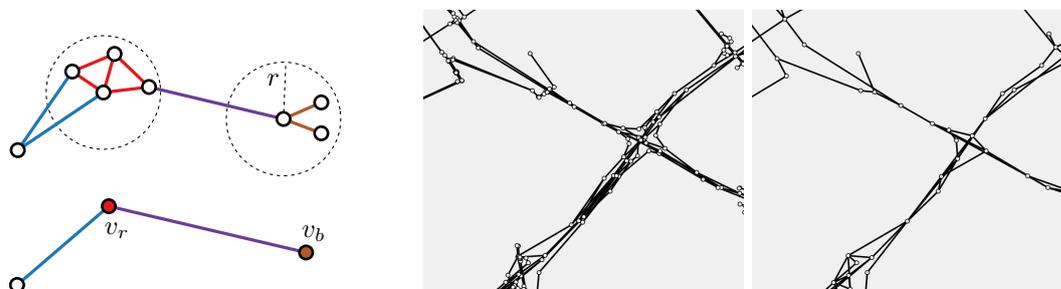
■ **Figure 3** (left) Replacing shallow turns; (middle) network before; (right) network after.

**Collapse dead ends** We first remove short paths in the network that do not increase the overall connectivity. Starting at a degree-1 vertex, we move along degree-2 vertices only to trace a visual “dead end” until we find a vertex that does not have degree 2. We compare its geometric length to a predefined parameter  $l_{max}$ , and collapse the path to this last vertex if its length falls below this threshold. We use  $l_{max} = 100m$  initially, and  $l_{max} = 1000m$  after the face-collapse operation. For coordination, we reroute any route along the collapsed edges to the endpoint that remains.

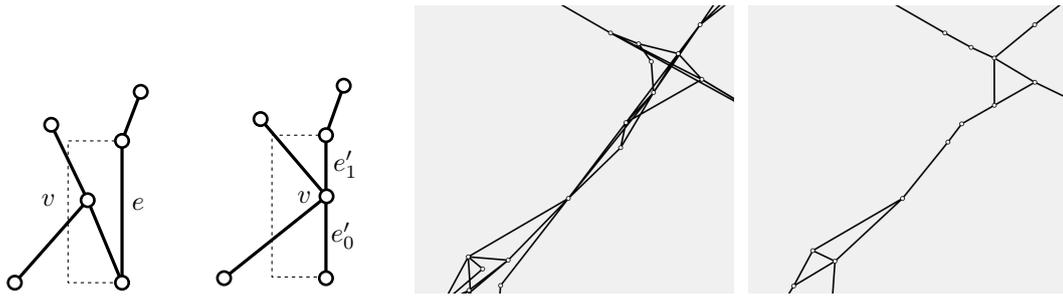
**Replace shallow turns** A detailed input network frequently contains small bends. The final schematization would remove such detail, but we perform this step early to simplify the merge and collapse operations to follow. For every degree-2 vertex, we consider the smallest angle between its incident edges. If this angle exceeds some predefined limit  $\beta$ , we replace the vertex and its two incident edges with a single edge; see Fig. 3. We use  $\beta = 150^\circ$  before and  $\beta = 140^\circ$  after the face-collapse step. For coordination, we reroute any route on one or both of the replaced edges  $e_0$  and  $e_1$  to the new edge  $e'$ .

**Merging vertices** Junctions in the road-network are too detailed for our schematic; ideally we represent them by a single vertex. To this end, we fix a radius  $r$  within which we merge vertices. For a vertex  $v$ , the merge operation for  $v$  merges all vertices within distance  $r$  of  $v$  (including  $v$  itself) to a single new vertex, placed at the centroid of the merged vertices (see Fig. 4). We iteratively merge vertices, prioritized by the number of vertices within radius  $r$ . We use  $r = 0.01D$  with  $D$  the length of the diagonal of the bounding box of the network. After merging faces, we use a larger radius of  $r = 0.03D$ .

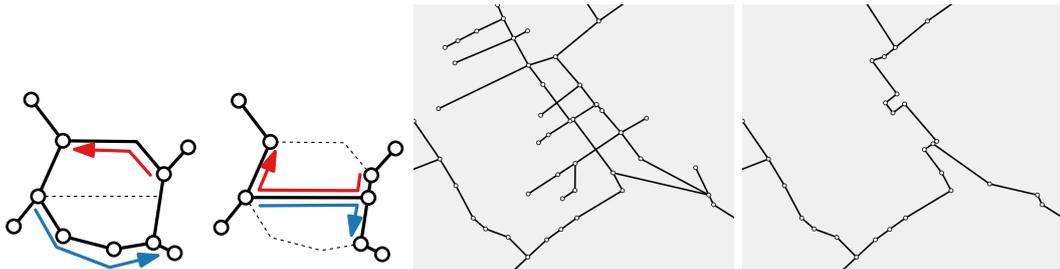
For coordination, edges inside the merge radius are mapped to the new vertex (e.g., red edges to  $v_r$  in figure). Edges between different new vertices or between a new vertex and an unmerged vertex are consolidated to new edges (purple and blue in figure).



■ **Figure 4** (left) Merging vertices; (middle) network before; (right) network after.



■ **Figure 5** (left) vertex-edge merging; (middle) network before; (right) network after.



■ **Figure 6** (left) Face collapse; (middle) network before; (right) network after.

**Merging vertices with edges** A vertex can be close to an edge, even though it is not close to its endpoints. In such cases, we merge vertices into nearby edges that are not incident to the vertex itself. We use the distance  $r$  as specified earlier, and try to collapse a vertex  $v$  to the closest non-incident edge  $e$  that is within distance  $r$ , where we demand that  $v$  lies in the slab spanned by  $e$  (see Fig. 5). For coordination, any route using the former edge  $e$  is now using the two (possibly new) edges  $e'_0$  and  $e'_1$  from  $v$  to the endpoints of  $e$ .

**Face collapse** We now consider the faces of the network and collapse them onto a single geometry, if they are “small”. Specifically, we collapse faces with an area of at most  $a_{max} = 0.01A$ , where  $A$  is the area of the bounding box of the network.

To collapse a small face  $F$ , we first collapse all interior degree-2 paths, independent of length. We then compute the minimal oriented bounding box of  $F$ , select its major axis, and cut  $F$  along this axis. In case of a non-convex face, we select the longest internal intersection between  $F$  and the axis (see Fig. 6). The cut introduces two cycles; we break both cycles by removing an edge or a path of degree-2 vertices from each; we can do so while maintaining the distances to the cut along the cycle for all higher-degree vertices. For coordination, we reroute any route that used the removed edges via the cut.

**Cleanup** After face merging, we repeat all earlier operations once more to clean up the geometry and prepare for computing the final schematization.

**Arc schematization** For our final schematization, we chose the arc schematization algorithm by van Dijk et al. [25]. This algorithm produces a low complexity schematization using (circular) arcs, while maintaining the topology of network. To encourage the use of straight lines over very shallow arcs, we increase the relative importance of straight lines by reducing the Fréchet distance for straight lines by a factor 0.3. Because this algorithm only removes vertices, coordination is straightforward, similar to replacing shallow turns.

**Time complexity.** The simplification operators run in roughly quadratic time with a straightforward implementation. The arc schematization algorithm dominates the operations to simplify the network. Thus, the time complexity is  $O(n^2 h \log n)$ , where  $h$  is the number

of vertices with degree higher than three in the input simplification and  $n$  the number of vertices, which is greatly reduced from the input at this point. Coordinating the changes between the road network and a trajectory  $T$  takes roughly  $O(|G||T|)$ .

## 6 Step 4: Detect bundles

**Desiderata.** We aim to detect bundles that are well-supported by the routes to provide insight into the overall mobility patterns. As such, we identify two desired properties of a bundle. First, it has to be supported by many routes. Specifically, we do not want a bundle representing a single trajectory (potentially an outlier), but rather the common behavior. Second, a bundle should be long in terms of (geometric) length. We aim to show mobility patterns that describe how vehicles move through the space. A long bundle is more descriptive of behavior than a short one; in the extreme case, a short bundle (even or especially if it is supported by many routes) may consist of one single edge (road segment), which does not help to communicate patterns beyond the existence of considerable local traffic. That is, we want to find long bundles that are supported by many routes.

As we aim to visualize not one but multiple bundles simultaneously, we further use three criteria to assess a set of bundles. First, we restrict our attention to *maximal* bundles only. That is, we consider only bundles to which we cannot add another route for its support, but specifically also not increase its length without having to reduce its support. Second, we generally want to see patterns of mobility that are *spatially diverse*: we prefer having bundles through different parts of the network, if the trajectories allow them. That is, we would like to avoid overlap between bundles as overlap communicates the same (local) behavior.

Third, a bundle fully contained within another may not provide much extra insight into mobility beyond showing a higher support for the contained route. We call a set of bundles *containment-free* if no bundle is a subroute of another. Note that containment-free bundles are not necessarily overlap-free and thus this is different from spatial diversity. Overlap-free bundles are containment-free, but we do not interpret spatial diversity as strict overlap-free.

In a containment-free set of maximal bundles each route may still support multiple bundles. If these are disjoint bundles, then we accept this as a bundle. However, as we are to eventually visualize the bundles, it may be misleading if well-supported bundles that share an edge are only well-supported because they share many trajectories. Thus, we choose to count the support for bundles in a disjoint manner: that is, routes through the overlap of bundles can be counted to support only one of these. We refer to this as the *disjoint* support.

Finally, note that we consider the network bidirectional, in the sense that each edge is present twice, once for each direction of travel. The considerations above should consider the direction of travel. That is, a bundle is directed and can, for example, only be supported by routes in the same direction. Two bundles that use the same set of edges, but travel in opposite direction, would hence be considered overlap-free.

**Related work.** Our bundling is closely related to finding groups of trajectories and finding representatives of trajectories, so-called *centers*. In the context of spatio-temporal data, grouping structures are used to find common patterns of mobility [5]. The groups that are constructed essentially are trajectories that move close together for a sufficient amount of time and with enough members in the group. Since we work with map-matched trajectories, we consider routes close when they share edges in the network, and we require bundles to have at least some minimum length.

Note that finding a group does not directly result in finding a representation for the behavior of the group. To represent the behavior of groups, a common approach is to cluster trajectories according to some metric, resulting in centers describing common behaviour for

the trajectories that are close to these centers under the used metric. This has been applied to the spatial component under different metrics and algorithmic approaches [6, 16, 18]; see [36] for a more comprehensive overview.

Since we work with map-matched routes, finding the bundles is similar to finding common substrings over a finite alphabet, where the alphabet is the set of edges and the routes form strings over this alphabet. Finding  $k$ -common substrings [4] is particularly related, since it demands a minimum number of matches  $k$ , similar to our high support requirement.

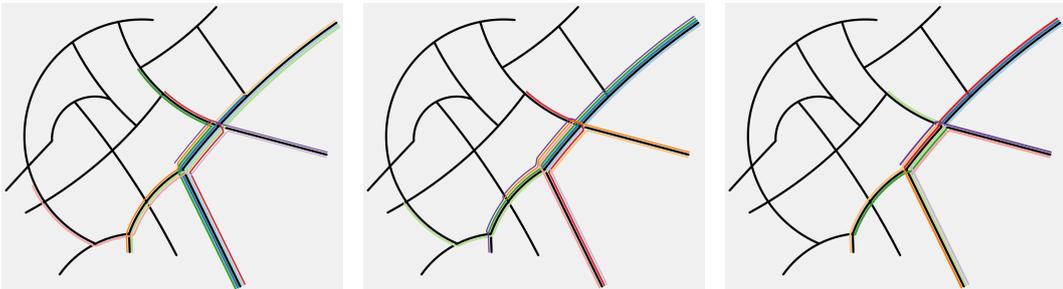
**Our implementation.** We search for a spatially diverse set of up to  $k$  bundles, where each bundle is a maximal bundle with a minimum length  $L_{\min}$  and minimum disjoint support  $S_{\min}$ . We construct this set by incrementally adding the “most informative” bundle.

To quantify this, we define the *importance* of a bundle in a way that allows a trade-off between length, disjoint support and spatial diversity. Specifically, the importance  $I(B)$  of a bundle  $B$  is defined as  $I(B) = L^p |S|^{1-p}$  for  $p \in [0, 1]$ , where  $L$  is the length of the bundle and  $S$  is the (disjoint) set of supporting routes. The rationale behind this is that, in the case of  $p = 0$  or  $p = 1$  we simply prioritize by total support or bundle length, respectively. However, if we set  $p = 0.5$ , then we prioritize the bundles by the total length of the route-set in its support. This allows a trade-off between length and disjoint support.

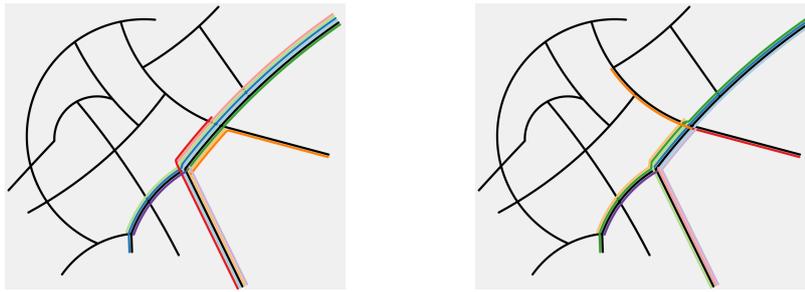
We define the length of a bundle  $B$  as  $\sum_{e \in B} \ell(e)$  for some function  $\ell$ . With  $\ell(e) = |e|$  (the Euclidean length) we promote long bundles directly. As we use disjoint support, there is already some preference for spatially diverse bundles, but we observe that this still leads to very similar bundles. To promote spatial diversity further, we may alter  $\ell$ . Specifically, we also allow for using  $\ell(e) = |e|/(1 + b(e))$  where  $b(e)$  is the number of selected bundles already using edge  $e$ . That is, conceptually, we “shrink” edges that have been used by other bundles. We thus refer to this setting as *shrunk edge lengths*.

We compute the most important bundle  $B$  by using a simple backtracking procedure. We then add  $B$  to our bundle set and remove its support from the complete set of routes. We repeat this process until  $k$  bundles have been found. If no bundle of sufficient length and support is found but we do not have  $k$  bundles yet, we halve  $S_{\min}$  and repeat. We halve  $S_{\min}$  at most twice, creating three *classes* of bundles (“thick”, “medium” and “thin” bundles).

Intuitively, focusing on bundle length (higher values of  $p$ ) is suitable for finding longer bundles that may have less support; useful to investigate longer mobility patterns. Reducing  $p$  focuses more on support and thus on patterns that are very frequent. In Fig. 7 (rendered according to our final step) we vary  $p$  and observe that reducing  $p$  (i.e., increasing importance of support) leads to less spatially diverse routes, but increases the bundle classes.



■ **Figure 7** Ten bundles for different importance schemes:  $p = 1$  (left),  $p = 0.5$  (middle) and  $p = 0$  (right). We used  $S_{\min} = 500$  and  $L_{\min} = 6000m$ , with shrunk edge lengths.



■ **Figure 8** Ten bundles for different approaches to spatial diversity: Euclidean edge lengths (left), shrunk edge lengths (right). We used  $S_{min} = 500$  and  $L_{min} = 6000m$  and  $p = 0.5$ .

Changing the edge lengths may encourage diversity, but may result in bundles of lower classes, as routes through existing bundles are “shorter” and thus less important. In Fig. 8 we compare the result using both our settings. Euclidean edge lengths show mostly similar routes, but we observe that shrunk edge lengths increase the spatial diversity. It does not avoid overlap, but it does reduce containment slightly (from 5 to 2 contained pairs).

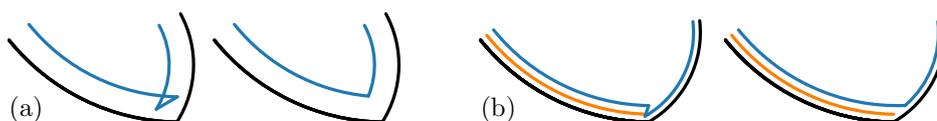
Computing one bundle takes  $O(|\mathcal{T}||\mathcal{G}|)$  time, where  $\mathcal{G}$  is the schematized network. This time includes the necessary changes to the information for computing the next bundle, so computing  $k$  bundles takes  $O(k|\mathcal{T}||\mathcal{G}|)$  time.

## 7 Step 5: Render schematic network with bundles

**Desiderata.** We now aim to jointly render the bundles and the schematic network, such that the bundles are clearly conveyed. For this, the bundles should be easily identified. Common practice is to use colors to identify the separate bundles, and in addition separate them visually. However, the bundles need to be rendered such that it is also easy to determine from what parts of the network they originate. Thus, rendering them in close spatial proximity to the associated edges would also be beneficial to the readability of the schematic. In addition, it should be possible to identify the (approximate) support for a bundle.

**Related work.** Rendering bundles in a network is similar to rendering the lines of a metro network, which is a well studied topic [32]. Common problems involve ordering lines on a common connection to avoid crossings and ensure good continuation at stations.

The offset rendering used in our implementation may want to avoid offsetting edges of a bundle with different distances, thus encouraging good continuation. If we allow gaps between bundles at an edge but disallow differences in offset within a single bundle, the problem is essentially to assign a layer to each bundle, minimizing the number of layers, such that two overlapping bundles use different layers. Even if the network would be a single cycle, this problem is NP-hard [11]; minimizing change is hence NP-hard as well.



■ **Figure 9** (a) A local self-intersection caused by offsetting and its resolution. (b) Local deviation from the direction of travel of the bundle and its resolution.

**Our implementation.** Our main approach is to render each bundle as a curve that is slightly offset from the network, such that they do not coincide with the network, nor each other. We visualize the direction of a bundle by offsetting its curves to a particular side of the network edges. As our example datasets are both in areas where cars drive on the right side of the road, we hence locally offset to the right. This scheme allows for identifying the direction of a bundle, without relying on other visual cues such as arrow heads.

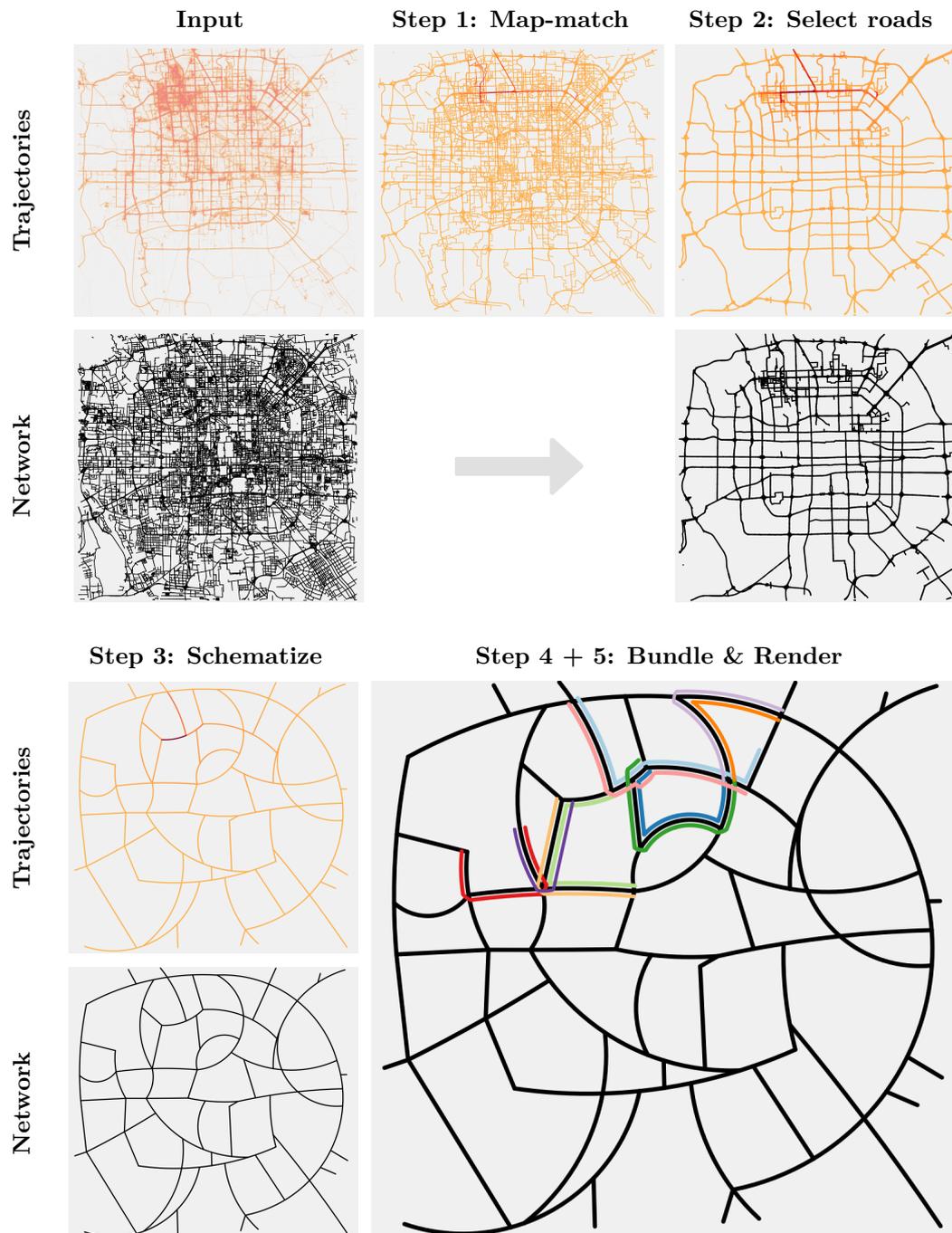
We lay out the bundles one at the time along the network. Each edge in the bundle gives an arc that is offset from the edge by a distance  $d(b + 1)$  for some constant  $d > 0$ , where  $b$  is the number bundles already placed at that edge. The bundle is now a sequence of arcs that do not quite connect correctly yet. We initially reconnect the arcs using straight segments. If this causes the curve to locally self-intersect (Fig. 9(a)) or cause small corners (Fig. 9(b)), directed opposite to the actual bundle direction, we simplify these artifacts as to achieve a simple curve that is always (roughly) directed in the direction of travel of the bundle. This operation takes  $O(l \log l)$  time, where  $l$  is the total complexity of the rendered bundles. Finally, we slightly smooth the connecting segments by reducing the arcs by a small distance and using the old endpoints as control points for a Bézier curve instead. We render the resulting curves using colors of the “10-class Paired” qualitative scheme from ColorBrewer (<https://colorbrewer2.org/>), and use a line thickness based on bundle class. By using classes instead of support, it is primarily aimed at separating main from secondary patterns.

## 8 Results

We implemented our proposed pipeline in C++, using MoveTK (<https://movetk.win.tue.nl/>) for trajectory processing and CGAL (<https://cgal.org/>) for geometric operations.

The HR dataset has been used to illustrate and discuss our pipeline throughout the paper. It covers the area of The Hague, the Netherlands. The network is obtained from OpenStreetMap (<https://www.openstreetmap.org/>) and has 60 277 vertices and 66 895 edges. In step 2, we use “primary” as the minimum road level for selection (<https://wiki.openstreetmap.org/wiki/Key:highway>). The GPS trajectories were provided by HERE Technologies (<https://www.here.com/>). As we are looking for patterns in mobility, we used only trajectories with a length of at least 10 000m that fall within the bounding box of the network. Trajectories partially within the bounding box were split and each part was treated as a separate trajectory. This left us with 3 795 trajectories as input.

The BJ dataset covers the metropolitan area of Beijing, China. The network is also obtained from OpenStreetMap and has 77 691 vertices and 132 167 edges. In step 2, we again use “primary” as the minimum road level for selection. The GPS trajectories originate from the open Geolife trajectory dataset by Microsoft [37], constrained to this region. We apply the same filtering step as for HR; this left us with 7 520 trajectories as input. The result of our pipeline is shown in Fig. 10. The Geolife dataset mainly contains trajectories obtained from taxi-drivers. We can clearly see in the heat-map of the dataset that there is a high concentration in the top-left, and in our schematic we see that relatively small, loop-like routes, capture a lot of the traffic in that region. Moreover, we see that most bundles occur in pairs, that is, two roughly identical bundles but in opposite directions. Our schematic map generally captures the outer ringroad, but struggles somewhat to capture the grid-like structure of the inner city. This is because the arc-based nature of our schematization algorithm is somewhat opposite to such structures. Future work may investigate hybrid approaches to schematizing such mixed networks of ring roads and non-grid-like structures with arcs, but grid-like structures with parallel segments.



■ **Figure 10** Our pipeline on the BJ data. The input trajectories are shown as a density map. For the map-matched routes, we use a orange to red scale to convey low to high traffic volume per edge. We compute the bundles in Step 4 with  $S_{min} = 150$ ,  $L_{min} = 8000m$ ,  $p = 0$  and shrunk edge lengths.

## 9 Discussion and future work

We propose a coordinated pipeline to create abstract visual summaries of mobility patterns in trajectory data. Our proof-of-concept implementation shows that the pipeline is feasible and can fully automatically compute such schematic maps. The advantage of a pipelined approach is that we may improve upon steps individually to improve the eventual result. Below, we reflect on our choices in the pipeline design, and discuss future work.

**Step order.** We could in principle also map-match (step 1) after selection (step 2). However, this would not allow for a data-driven approach for selection. Moreover, this forces all traffic to the selected roads, which may hide information on traffic that does not follow the selected (major) roads. For HR, the data-driven selection seems to not have made a large impact on the selected network; for BJ more extensive parts of the network were added because of the data-driven selection. We leave to future work to investigate whether inversion of these steps is able to provide meaningful visualizations.

Similarly, we may wonder whether we would want to detect bundles (step 4) before schematization (step 3). As schematization reduces the network complexity, it is more efficient to do afterwards. It may distort distances, but we can keep track of the original distances if desired – note that aggregation in step 3 does not make each route in the schematic network map to precisely one route in the original network. Another reason supporting our choice for the given order is that schematization may further aggregate dense areas of the network. By bundling afterwards, the support for such bundles grows since they are effectively representing more traffic that generally traverses the dense area in roughly the same way. We believe that our choice helps in promoting spatial diversity, as dense areas with low traffic per road may reduce to a single road with higher traffic. In light of our very spatially uneven datasets, this seems desirable. However, we leave the full exploration of the impact of this choice as future work.

**Augmenting the schematic map.** We split the map-matched trajectories according to whether or not the route is on the selected network. This leaves us with parts of the trajectories that go through unselected parts of the network and are thus dropped from the schematic map. We intend to explore ways of visualizing these dropped subroutes to provide information on the traffic not part of the selected network. On a computational level, an approach we see for this is to track these subroutes relative to the faces of an embedding of the network. This, however, demands that we meticulously keep track of what happens to these faces during the simplification stages. But it also requires visual design: what do we want to show of these dropped routes, and how does that combine with the shown bundles?

While our mapping between simplifications is discrete in nature, an interesting direction of research would be to extend this to continuous mappings, where routes are also allowed to start in the middle of edges. An appropriate map-matcher should also be selected, since the Fréchet map-matching approach maps only to full edges, though we expect the overall impact of allowing continuous routes here to be minor, as it is performed on the original, detailed network. The primary question is how we can alter the schematization step to allow for high-quality continuous mapping through aggregation and simplification of the network.

**Using the schematic map.** In our proposed pipeline, we do not incorporate the time component of the input trajectories explicitly. Given that the schematic shows strongly aggregated data in a concise way, we can easily use our approach to show small-multiples for different time frames in the data set. This leaves open the question what the best selection method of the road-network is in this case, which we defer to future work.

Our method hides any of the effects of sampling and noise in the data, as well as the deformation and aggregation that occurs in our pipeline, which may be unintuitive to end users. Though the schematic appearance aims to implicitly convey such information, it may be communicated more explicitly with additional uncertainty visualization, albeit at the cost of added visual complexity.

The final schematization is strongly influenced by the selected parameters in steps of our pipeline. We scale parameters by a typical size (e.g. bounding box diagonal or area) to be able to assign parameters independent of scale. Nevertheless, visualizing the impact of the parameters on the end result could help users pick appropriate values for their use cases.

---

## References

- 1 N. Adrienko and G. Adrienko. Spatial generalization and aggregation of massive movement data. *IEEE TVCG*, 17(2):205–219, 2010. doi:10.1109/TVCG.2010.44.
- 2 H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262–283, 2003. doi:10.1016/S0196-6774(03)00085-3.
- 3 G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao. Visual Analytics of Mobility and Transportation: State of the Art and Further Research Directions. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2232–2249, 2017. doi:10.1109/TITS.2017.2683539.
- 4 M. Arnold and E. Ohlebusch. Linear time algorithms for generalizations of the longest common substring problem. *Algorithmica*, 60(4):806–818, 2011. doi:10.1007/s00453-009-9369-1.
- 5 K. Buchin, M. Buchin, M. van Kreveld, B. Speckmann, and F. Staals. Trajectory grouping structure. In *WADS*, pages 219–230, 2013. doi:10.1007/978-3-642-40104-6\_19.
- 6 K. Buchin, A. Driemel, J. Gudmundsson, M. Horton, I. Kostitsyna, M. Löffler, and M. Struijs. Approximating  $(k, l)$ -center clustering for curves. In *Proc. SODA*, pages 2922–2938, 2019. doi:10.1137/1.9781611975482.181.
- 7 S. Cabello, M. de Berg, and M. van Kreveld. Schematization of networks. *Computational Geometry*, 30(3):223–238, 2005. doi:10.1016/j.comgeo.2004.11.002.
- 8 P. Chao, Y. Xu, W. Hua, and X. Zhou. A survey on map-matching algorithms. In *ADC*, pages 121–133, 2020. doi:10.1007/978-3-030-39469-1\_10.
- 9 J. Chen, Y. Hu, Z. Li, R. Zhao, and L. Meng. Selective omission of road features based on mesh density for automatic map generalization. *IJGIS*, 23(8):1013–1032, 2009. doi:10.1080/13658810802070730.
- 10 W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015. doi:10.1109/TITS.2015.2436897.
- 11 M. Garey, D. Johnson, G. Miller, and C. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic Discrete Methods*, 1(2):216–227, 1980. doi:10.1137/0601025.
- 12 H. Imai and M. Iri. Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics, and Image Processing*, 36(1):31–41, 1986. doi:10.1016/S0734-189X(86)80027-5.
- 13 M.-J. Kraak. The Space-Time cube revisited from a Geovisualization perspective. In *Proc. ICC*, pages 1988–1996, 2003.
- 14 L. Kulik, M. Duckham, and M. Egenhofer. Ontology-driven map generalization. *Journal of Visual Languages & Computing*, 16(3):245–267, 2005. doi:10.1016/j.jv1c.2005.02.001.
- 15 O. Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *Proc. IEEE PacificVis*, pages 171–178, 2011. doi:10.1109/PACIFICVIS.2011.5742387.
- 16 J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proc. 2007 ACM SIGMOD*, pages 593–604, 2007. doi:10.1145/1247480.1247546.
- 17 W. Meulemans. *Similarity Measures and Algorithms for Cartographic Schematization*. PhD thesis, TU Eindhoven, 2014. doi:10.6100/ir777493.

- 18 A. Nath and E. Taylor. k-Median clustering under discrete Fréchet and Hausdorff distances, 2020. [arXiv:2004.00722](https://arxiv.org/abs/2004.00722).
- 19 R. Scheepens, N. Willems, H. van de Wetering, and J. van Wijk. Interactive visualization of multivariate trajectory data with density maps. In *IEEE PacificVis*, pages 147–154, 2011. doi:10.1109/PACIFICVIS.2011.5742384.
- 20 R. Šuba, M. Meijers, and P. van Oosterom. Continuous road network generalization throughout all scales. *ISPRS Journal of Geo-Information*, 5(8):145, 2016. doi:10.3390/ijgi5080145.
- 21 R. Thomson. The ‘stroke’ concept in geographic network generalization and analysis. In *Progress in Spatial Data Handling*, pages 681–697. Springer, 2006. doi:10.1007/3-540-35589-8\_43.
- 22 C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE TVCG*, 18(12):2565–2574, 2012. doi:10.1109/TVCG.2012.265.
- 23 M. van de Kerkhof, I. Kostitsyna, M. van Kreveld, M. Löffler, and T. Ophelders. Route-preserving road network generalization. In *Proc. ACM SIGSPATIAL*, pages 381–384, 2020. doi:10.1145/3397536.3422234.
- 24 S. van den Elzen and J. van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE TVCG*, 20(12):2310–2319, 2014. doi:10.1109/TVCG.2014.2346441.
- 25 T. van Dijk, A. van Goethem, J.-H. Haunert, W. Meulemans, and B. Speckmann. Map schematization with circular arcs. In *GIScience*, pages 1–17, 2014. doi:10.1007/978-3-319-11593-1\_1.
- 26 A. van Goethem, W. Meulemans, A. Reimer, H. Haverkort, and B. Speckmann. Topologically safe curved schematisation. *The Cartographic Journal*, 50(3):276–285, 2013. doi:10.1179/1743277413Y.0000000066.
- 27 M. van Kreveld. Smooth generalization for continuous zooming. In *IGU*, pages 2180–2185, 2001. URL: [https://icaci.org/files/documents/ICC\\_proceedings/ICC2001/icc2001/file/f13042.pdf](https://icaci.org/files/documents/ICC_proceedings/ICC2001/icc2001/file/f13042.pdf).
- 28 P. van Oosterom. Variable-scale topological data structures suitable for progressive data transfer: The GAP-face tree and GAP-edge forest. *Cartography and Geographic Information Science*, 32(4):331–346, 2005. doi:10.1559/152304005775194782.
- 29 R. Weiss and R. Weibel. Road network selection for small-scale maps using an improved centrality-based algorithm. *Journal of Spatial Information Science*, 2014(9):71–99, 2014. doi:10.5311/JOSIS.2014.9.166.
- 30 A. Wolff. Drawing subway maps: A survey. *Informatik-Forschung und Entwicklung*, 22(1):23–44, 2007. doi:10.1007/s00450-007-0036-y.
- 31 J. Wood, J. Dykes, and A. Slingsby. Visualisation of Origins, Destinations and Flows with OD maps. *The Cartographic Journal*, 47(2):117–129, 2010. doi:10.1179/000870410X12658023467367.
- 32 H.-Y. Wu, B. Niedermann, S. Takahashi, and M. Nöllenburg. A survey on computing schematic network maps: The challenge to interactivity. In *Proc. Schematic Mapping Workshop*, 2019. URL: <https://www.cg.tuwien.ac.at/research/publications/2019/wu-2019-smw/>.
- 33 B. Yang, X. Luan, and Q. Li. Generating hierarchical strokes from urban street networks based on spatial pattern recognition. *IJGIS*, 25(12):2025–2050, 2011. doi:10.1080/13658816.2011.570270.
- 34 Y. Yang, T. Dwyer, S. Goodwin, and K. Marriott. Many-to-many geographically-embedded flow visualisation: An evaluation. *IEEE TVCG*, 23(1):411–420, 2016. doi:10.1109/TVCG.2016.2598885.
- 35 W. Yu, Y. Zhang, T. Ai, Q. Guan, Z. Chen, and H. Li. Road network generalization considering traffic flow patterns. *IJGIS*, 34(1):119–149, 2020. doi:10.1080/13658816.2019.1650936.
- 36 G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang. A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, 47(1):123–144, 2017. doi:10.1007/s10462-016-9477-7.
- 37 Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. WWW*, pages 791–800, 2009. doi:10.1145/1526709.1526816.

# Harmonious Simplification of Isolines

Arthur van Goethem ✉

Eindhoven University of Technology, The Netherlands

Wouter Meulemans ✉ 

Eindhoven University of Technology, The Netherlands

Andreas Reimer ✉

Eindhoven University of Technology, The Netherlands

Bettina Speckmann ✉ 

Eindhoven University of Technology, The Netherlands

---

## Abstract

Current techniques for simplification focus on reducing complexity while maintaining the geometric similarity to the input. For isolines that jointly describe a scalar field, however, we postulate that geometric similarity of each isoline separately is not sufficient. Rather, we need to maintain the *harmony* between these isolines to make them visually relate and describe the structures of the underlying terrain. Based on principles of manual cartography, we propose an algorithm for simplifying isolines while considering harmony explicitly. Our preliminary visual and quantitative results suggest that our algorithm is effective.

**2012 ACM Subject Classification** Information systems → Geographic information systems; Theory of computation → Computational geometry

**Keywords and phrases** Simplification, isolines, harmony

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.8

**Supplementary Material** *Software (Source Code)*: <https://github.com/tue-alga/harmonious-simplification>

**Acknowledgements** DEMs provided by the Byrd Polar and Climate Research Center and the Polar Geospatial Center under NSF-OPP awards 1543501, 1810976, 1542736, 1559691, 1043681, 1541332, 0753663, 1548562, 1238993 and NASA award NNX10AN61G. Computer time provided through a Blue Waters Innovation Initiative. DEMs produced using data from DigitalGlobe, Inc.

## 1 Introduction

Isolines are among the most common and effective cartographic techniques for visualizing scalar field-type data, such as terrain, precipitation, air pressure or, more abstractly, travel times. Isolines as a visualization method share and mirror an aspect of fields themselves: only the multitude of isolines signifies the phenomenon and the objects that it comprises. A single spot height is insufficient to understand the lay of the land, and equally insufficient is the individual isoline. It follows naturally, that the visual and the geometric relationships between individual, pairs, and groups of isolines are essential to their cartographic effectiveness.

Geographic data need to be represented at different scales for specific tasks. The processes that transform the visualization of geographic datasets due to scale-changes are subsumed under the term generalization. Our approach aims to partially capture interdependencies in the form of “harmony” of isolines while generalizing them. Visual harmony as a concept alludes to neighboring isolines behaving in a similar manner where such behavior is warranted. In cartographic practice, questions of harmony used to be addressed by the human cartographer. Nowadays isolines are usually generated automatically by interpolation processes from scalar fields, with the most common example being Digital Terrain Models (DTM) [32]. As most progress in automated generalization has been made for topographic maps, research in automatization of isolines so far has concentrated on the generalization of terrains, with systems of contour lines being one of the major outputs [16].



© Arthur van Goethem, Wouter Meulemans, Andreas Reimer, and Bettina Speckmann; licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 8; pp. 8:1–8:16

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are two main advantages to generalizing isolines instead of resampling a scalar-field (usually a DTM) and then interpolate fresh isolines: (1) the ability to *directly* keep relations intact within a multi-resolution database, and (2) the avoidance of visual jumps and resampling artifacts. Consider, for example, a river which must always run through the deepest part of a valley and as such must be tied to the isolines. When resampling a DTM and generalizing the drainage polylines independently, the generalized rivers might run over the slopes of the freshly derived contour line system [25]. Hence, until a few years ago, many national mapping agencies either kept their manually generalized isolines for their existing scale ranges and did not automatically derive them for production of topographic maps, or employed sophisticated techniques to tie freshly generated contour lines to pertinent database objects and/or the DTM (such as the GAEL-model for Object-Field Relation Preservation [9]). Such conflation methods are not applicable to most other scalar fields due to lack of ancillary data. They have also been out of reach for most GIS users, although some open-source plugins for DTM-based tasks have been released recently [25, 30].

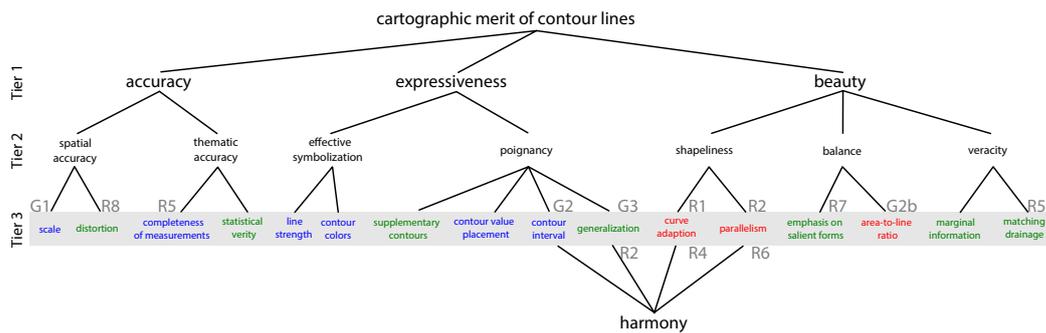
**Results and organization.** In this paper we propose an algorithm to simplify a whole system of vector isolines while considering harmony explicitly. In Section 1.1 below, we present a framework which allows us to consider harmony as an element of cartographic merit of systems of contour lines. We describe our harmonious simplification algorithm in Section 2 and present experimental results in Section 3. Our examples focus on contour lines, but we note that our algorithm works with any isoline vector input.

Our method allows us to apply and refine many heretofore neglected design rules (such as those related to harmony) for isoline systems. It also naturally supports animated morphs between scales without loss of connections. Last but not least, since our algorithm is targeted towards vector data, it allows us to apply locally adaptive generalization.

## 1.1 Harmony

While the word harmony has a wide range of meanings, we are concerned only with harmony in relation to isolines. On first sight, the harmonious visual interaction between isolines looks like a lofty or imprecise concept. However, the notion that a pleasing and informative repetition that goes beyond mere parallelism is a crucial design goal, permeates the literature. In Swiss cartographic debates of the early 20th century, harmony was explicitly named as an intermediary design goal, which directly influenced academic and production mountain and terrain cartography [27]. To move from the qualitative and aesthetic concept of harmony to actionable requirements, we derive the pertinent design rules from cartographic examples and cartographic literature. From those design rules, we move on to measurable qualities.

The basic idea of our approach of conceptualizing harmony is a modification of the process used by Reimer [24] for deriving and modeling harmony for chorematic diagrams. Reimer follows Goldman [13] and observes that the general merit of a map can be operationalized by linking objective, measurable qualities to aesthetic qualities of varying abstractness. This is done hierarchically: the general merit of a map is a function of its *accuracy*, *expressiveness*, and *beauty*, with *originality* an optional addition that may well be zero, and is indeed zero in our case. The first three are the first tier aesthetic qualities. To each of them, Reimer [24] links a number of second tier aesthetic qualities: more precise, but still not directly measurable. The third hierarchical tier then links measurable qualities to the second tier. While the first and second tier are capturing evaluative concepts which apply to every cartographic expression, the third tier is specific to a type of map or map element. That is,



■ **Figure 1** Hierarchical structure of the cartographic merit for isolines, measurable properties in Tier 3. Imhof [18] provides either values or examples for all blue properties, the cartographic literature provides further guidance for green properties. There are no established values for red properties. We define harmony as a combination of four Tier 3 properties: contour interval, generalization, curve adaptation, and parallelism.

the measurable qualities as formalized in the third tier apply to a specific map type, which are isolines in our case (see Fig. 1). Once we have formalized the third tier for isolines, we can define harmony as a combination of third tier elements.

**Tier 3: measurable qualities for isolines.** For the case of contour lines, the accepted master both on aesthetic as well as scientific design rules is Eduard Imhof [18]. The three basic overarching dimensions of Imhof’s conceptualization of a good contour line system arguably match directly to Goldman’s top tier: “Auswertbarkeit” corresponds to accuracy, “Lesbarkeit” corresponds to expressiveness, and “Empfinden” corresponds to beauty. Imhof operationalizes these dimensions into concrete global qualities, namely:

- G1** Scale-appropriate terrain data.
- G2** Contour interval matching the target scale and depicted terrain.
  - a Round numbers for easy counting preferred.
  - b Area between isolines shall be neither too big nor too small.
  - c Harmony between all elements of the system.
- G3** Easy readability, including generalization, where needed.

As we are interested in generalizing isolines and assuming a stable **G1** and **G2** setting, we are especially interested in his comments on **G2c** and **G3**. Below we summarize Imhof’s design rules for small-scale contour line generalization [18]:

- R1** For each small area of terrain, design the whole line system together.
- R2** Smooth over small jagged parts in a single line if it is not reflected in neighboring lines.
- R3** Neighboring contours should never touch. Exceptions for alpine regions (cliffs, karren).
- R4** Shapes should reflect reality; sometimes with pointy and jagged parts, sometimes gentle curves, but neither to the exclusion of the other.
- R5** Use ancillary data like terrain crests or breaks of slope when drawing the contour lines.
- R6** If a small form cuts across several contours, it should be treated as a whole. Either retain and draw it everywhere or remove it.
- R7.1** Retain prominent, convex, terrain features which are much more visible in-situ than hollow forms.
- R7.2** Emphasize important hollow forms, widening them for legibility if necessary.
- R8** Minimize geometry changes when following the other rules.

For **R3** Imhof specifically highlights that the whole line system must be brought under control and carefully adjusted, not just the overlapping pairs. This is a recurring theme with Imhof's design rules which also appears in label placement. We populated Tier 3 with the measurable properties discussed by Imhof, either in the form of rules or in the form of examples. We added statistical verity as well as marginal information.

Imhof does not necessarily provide evaluation functions or even parts thereof together with his rules. An example would be **G3b**, namely the ratio of area to line, which must be somehow balanced, but no further information is provided. In Figure 1 we highlighted those Tier 3 properties in blue for which Imhof gives sufficient guidance. We further highlighted those properties in green for which sufficient guidance can be found in the cartographic literature. There remain three properties for which Imhof presents rules, but not sufficient guidance, namely curve adaption, parallelism, and area to line ratio.

We are now ready to define *harmony of isolines*: a melange of elements of poignancy and shapeliness. Following the hierarchy, harmony is potentially measurable by evaluation functions for generalization solutions, curve adoption and parallelism.

## 1.2 Related work

A very good overview until 2014 of automated terrain generalization, of which contour line generation is a subset, can be found in [16]. Since then, recent work was done, for example, on isobath generalization [15, 28], continuous generalization via morphing [10], improving conflation of raster DTMs with contour lines [25, 30] and generation of supplementary contour lines [26]. These examples highlight a growing interest into automated sea-chart production, online mapping and continued support for NMA-demand via ancillary data-driven processes based on DTMs for multiple resolution databases. Isobaths are often considered a special case, due to the peculiar needs for safety constraints [15] as well as being not entirely treated as representations of a continuous scalar field but rather discrete maritime objects. Although direct generalization of isobaths from other isobaths in the vector realm is more common than in current dry-land cartography, they are thus not too much concerned with how the individual line shapes interact. Current morphing approaches such as [10] use a vertex matching between origin and target scale and then interpolate on that matching, with few regards to cartographic design rules.

Moreover, most proposed techniques for the simplification of isolines (e.g., [10, 20]) focus on individual isolines without explicitly considering the relationship between adjacent isolines. Recently, van Goethem et al. [11] proposed a technique for simultaneous simplification of isolines with a focus on parallelism. They strengthen the parallelism between adjacent isolines while simplifying the input. However, the relative positions of the vertices on adjacent isolines is not explicitly taken into account, thereby addressing but one facet of harmony.

In recent years several automated simplification approaches have been proposed that allow the introduction of new vertex locations to more closely approximate the original input at lower complexity simplifications. The potential new vertex positions are usually limited, for example using criteria such as maintaining local topology [7, 12], interpolation between scales [10], or using an underlying grid [23]. An alternative criterion that is often used is area preservation (e.g., [6, 12, 19, 22, 29, 31]). Area preservation mandates that the area covered by the simplified feature is exactly equal to the area covered by the original feature. Area preservation ensures that features can not arbitrarily increase or decrease in size during the simplification process and can be used to significantly reduce the search space for new vertex positions. These criteria are often combined together with a standard step to introduce new vertices. One of the more prominent techniques used for this is the *edge*

*collapse* (e.g. [19, 31]). In an edge collapse a sequence of three consecutive edges is replaced by two new edges, implicitly “collapsing” the middle edge to a vertex. Our proposed technique follows these principles of area-preserving edge collapses, but performs such operations in parallel, so as to promote harmony.

Computing an optimal simplification under the criteria of area preservation is a hard computational problem. Bose et al. [4] showed that even when solely considering an  $x$ -monotone path computing a simplification of  $k$  links that is area preserving is NP-hard. Löffler and Meulemans [21] showed that even when the resulting area-preserving simplification is restricted to only having (horizontal and vertical) edges on an orthogonal grid the problem is NP-hard. Similar results appear when considering simplification in a network. Estkowski and Mitchell [8] have shown that minimizing the number of (degree-2) vertices in a network subject to a given error bound while preventing intersections is NP-hard to approximate to within a factor  $n^{\frac{1}{5}-\delta}$ ,  $\delta > 0$ . The results discussed above underpin the necessity for efficacious heuristics, such as our proposed method, for computing (harmonious) simplifications.

## 2 Harmonious simplification algorithm

We present a simple, efficient algorithm to compute the harmonious simplification of a sequence of polygonal lines. We assume the polylines represent a consistent part of the terrain (a common snippet of isolines). We note that our techniques readily apply to polygons as well, and can be generalized to more complex arrangements of isolines. The exact steps of such generalization are out of scope for this paper and will be addressed in future work.

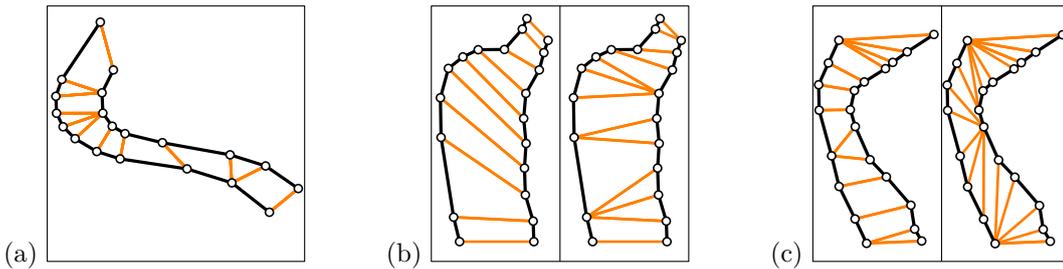
Our algorithm consists of two main phases: (1) preprocess the polylines to identify so-called *slope ladders*; (2) repeatedly *collapse* slope ladders to reduce complexity while maintaining harmony. Our initial input is a sequence of polylines  $\mathcal{P} = \langle P_1, \dots, P_k \rangle$  where each  $P_h = \langle v_{h,1}, \dots, v_{h,n_h} \rangle$  is a polygonal line defined by  $n_h$  vertices in the plane. Each polyline represents an isoline and they are ordered by the height they represent, thus consecutive isolines are also consecutive in the order. We assume that all polylines are given in a consistent orientation; it is straight-forward to ensure this assumption is always met. Because the polylines represent isolines, they do not intersect each other nor self-intersect.

### 2.1 Preprocessing: deriving slope ladders

To maintain harmony during simplification we first need to detect the harmony that is present in the input. We determine a *matching* between the vertices of every consecutive pair of isolines and then combine these into *slope ladders* stretching multiple isolines.

**Matching.** The matching between the isolines describes how the adjacent isolines are related to each other, particularly which pairs of vertices are considered “nearby”, and thus implicitly describes the terrain represented by the isolines. Such a matching could also be derived from slope information if available. However, lacking such information, several algorithms exist that can compute such a matching based on the geometry of the isolines alone. In this paper we focus on such geometric algorithms, and particularly consider Dynamic Time Warping (DTW) [3] and the (discrete) Fréchet distance [2]. In this paper, we use Fréchet distance to refer to the discrete variant, rather than the continuous variant.

We require that a matching between polylines  $P_h$  and  $P_{h+1}$  maps every vertex of  $P_h$  to at least one vertex of  $P_{h+1}$  and each vertex of  $P_{h+1}$  to at least one vertex of  $P_h$  such that the matching is *monotone*. That is, if vertex  $v_{h,i}$  is matched to  $v_{h+1,j}$  then every vertex  $v_{h,i'}$ , with  $i' > i$ , can only be matched to vertices  $v_{h+1,j'}$  where  $j' \geq j$ , and similarly every



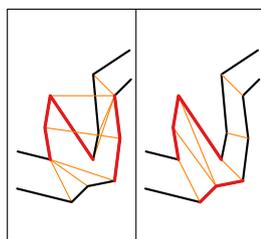
■ **Figure 2** (a) The polyline matching should be monotone. (b) The DTW matching (left) minimizes the number of matched pairs instead of preserving locality (right). (c) Matchings with the same Fréchet distance; the left matching is locally correct.

vertex  $v_{h+1,j''}$ , with  $j'' > j$  must be matched to vertices  $v_{h,i''}$ , where  $i'' \geq i$ . Intuitively this gives rise to a sequence of matchings that have a consistent order along both polylines (see Fig. 2(a)). Both DTW and the Fréchet distance adhere to these requirements.

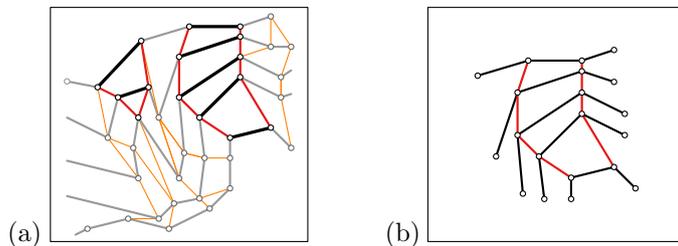
As we assume the terrain to be locally consistent with the isolines, matched vertices should generally be nearby. While both the DTW and the Fréchet distance capture this locality to some degree, both are not completely satisfactory. Particularly the DTW minimizes the *sum* of distances between all matched vertices. Therefore the DTW tends to prefer fewer pairs even if the pairs of vertices are not close together (see Fig. 2(b)). In contrast the Fréchet distance focuses purely on the *maximum* distance between any pair of matched vertices (the *bottleneck*). However, no further distinction is made between all matching that minimize the bottleneck (see Fig. 2(c)). This may lead to matchings that may not necessarily describe similarity well [5]. To overcome this issue, Buchin et al. introduced locally correct Fréchet matchings (LCFM) [5]. These matchings recursively enforce the bottleneck, ensuring the lowest distances between matched pairs are achieved for all pairs. Such a locally correct (discrete) Fréchet matching can be computed in the same time as the Fréchet distance.

We observe, however, that using the Euclidean distance for the matching may still lead to counterintuitive results as the matching line is exterior to the area bounded by the two isolines (see Fig. 3). To prevent such counterintuitive matchings, we use the geodesic distance in the bounded area instead. We abbreviate the matchings as ELCFM and GLCFM to distinguish the Euclidean and Geodesic variants, and again use only the discrete variants.

**Slope ladders.** We combine the pairwise matchings between consecutive polylines to obtain a *slope skeleton* of the hillside described by the isolines. The slope skeleton relates the underlying geometry across all isolines (**R1**). For each edge of a polyline we distinguish



■ **Figure 3** Locally correct Fréchet matchings using the Euclidean distance (left) and using the geodesic distance (right).



■ **Figure 4** (a) Two examples of slope ladders in this skeleton, one of complexity 2 and one of complexity 5. (b) The rungs of the slope ladder (purple) and the associated sequences of four vertices.

two cases. Either both endpoints of the edge are mapped to the same vertex on the adjacent polyline (creating a triangle) or they map to two consecutive vertices (creating a quadrilateral).

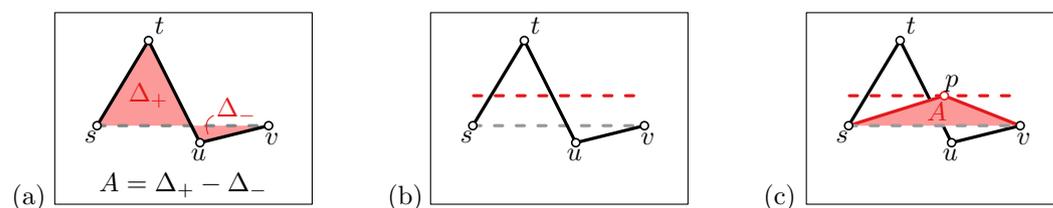
We maximally join all quadrilaterals and triangles that share a polyline edge into what we call a *slope ladder*: a sequence of quadrilaterals, possibly capped with triangles (see Fig. 4(a)). A slope ladder can intuitively be thought of as describing a sloped face of the hillside. The polyline edges in a slope ladder we refer to as the *rungs* of the ladder (see Fig. 4(b)). Each rung is associated with four vertices  $\langle s, t, u, v \rangle$ : the endpoints of the previous edge, the rung, and the next edge of the polyline. The complexity of a slope ladder is its number of rungs.

The slope skeleton naturally, uniquely, and fully decomposes into slope ladders: every polyline edge is a rung in precisely one slope ladder. We can straightforwardly compute the slope ladders by starting a new slope ladder from an edge which is not assigned to a slope ladder yet, and using the matching with adjacent polylines to find new edges of the ladder, until the matching matches both endpoints to the same vertex.

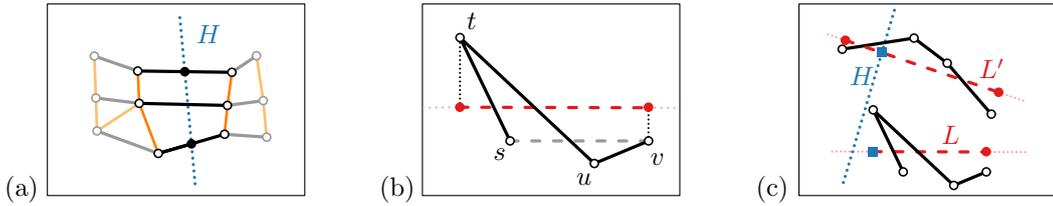
## 2.2 Iterative reduction: a single collapse

We simplify the polylines while ensuring that matched vertices are either maintained or removed as a whole (R6). This explicitly enforces harmony as we simplify the isolines. We do so by iteratively *collapsing* slope ladders. To further strengthen harmony we place new vertices such that they form a more visually striking cross-isoline feature. Particularly we ensure that newly placed vertices are collinear.

**Area-preserving collapses.** To collapse a slope ladder, we replace each of its rungs by a single vertex. Consider a rung  $tu$  with associated vertices  $\langle s, t, u, v \rangle$  of polyline  $P_h$ . A collapse removes  $t$  and  $u$  from  $P_h$  and inserts in their place a single new vertex  $p$  (see Fig. 5(a,c)). The new vertex may be placed freely and need not be restricted to the input vertices. In line with other algorithms, we require area preservation for each isoline, such that the area between two isolines remains the same. Let  $A$  be the signed area of the region enclosed by the segment  $sv$  and the segments between vertices  $s$  to  $v$  in reverse order (see Fig 5(a)). This is the area that would be lost (respectively gained) between the adjacent isolines. We place the new vertex  $p$  such that an equivalent area is gained (respectively lost). As  $spv$  forms a triangle and  $sv$  is a fixed base, the height of the triangle with respect to this base is  $h = \frac{2A}{|sv|}$ . Thus the placement of  $p$  is limited to a line parallel to the segment  $sv$  at height  $h$  (see Fig 5(b)). We refer to it as the *area-equivalence line* for rung  $tu$ : any  $p$  on this line ensures exact area preservation (see Fig. 5(c)).



■ **Figure 5** (a) Area contributed by the original rung and its adjacent edges. (b) The area-equivalence line. (c) Any position of the area-equivalence line creates a triangle that contributes the same area as before. (Suboptimal position selected for visualization purposes.)



■ **Figure 6** (a) *Harmony line*  $H$  for a slope ladder. (b) *Span* (dark red dashed) of a single rung on the area-equivalence line. (c) The new position (blue boxes) is the intersection of  $H$  and  $L$  ( $/L'$ ) snapped to the nearest point on the *span*. (Arbitrary harmony line for visualization purposes.)

**Harmonious candidate positions.** To promote harmony we simultaneously select the position of the new vertices for all rungs (**R1**, **R6**). If there is only one rung in the ladder then we simply use the area-equivalence line and sample it to find the optimal location. In the following we assume there are at least two rungs. We have two criteria for the placement of the new vertices: (1) the position should reflect the underlying geometry well; (2) the positions of the new vertices should be harmonious. We enforce the second criterion by requiring that all new points are collinear. Given this constraint we then optimize for the first criterion (**R4**, **R8**).

As we require the vertices to be collinear, the only freedom left is to fix the line on which the new vertices will be placed. This line should ideally be perpendicular to the isolines themselves, to ensure good alignment of the incident edges for the new vertices. To this end we construct an initial *harmony line*  $H$  through the midpoints of the first and last rung in the slope ladder (see Fig. 6(a)). We assume the initial harmony line  $H$  intersects the area-equivalence line of each rung; if it does not, we do not allow collapsing this slope ladder.

The placement of the harmony line directly defines the placement of all vertices. Consider a single rung. Let  $\langle s, t, u, v \rangle$  be the consecutive four vertices along the associated isoline that are the endpoints of the rung, the previous edge and next edge. Let  $L$  be the area-equivalence line belonging to the rung. The *span* of rung  $tu$  with respect to  $L$  is the maximum interval on  $L$  spanned by a pair of vertices from  $\langle s, t, u, v \rangle$  (see Fig. 6(b)). If the intersection of  $H$  with  $L$  lies inside the span of the rung, then the position  $p$  at which the new vertex will be placed is the intersection of  $H$  and  $L$ , otherwise  $p$  is the closest endpoint of the span to the intersection of  $H$  and  $L$  (see Fig. 6(c)).

Directly using the initial harmony line is too restrictive. To improve flexibility we allow the harmony line to be offset (perpendicular to its orientation) to find an optimal placement of the new vertices. For each offset of  $H$  we compute the new vertex locations and how well they approximate the original isolines. As a measure of the induced error per rung we use the directed Hausdorff distance from the newly inserted two edges to the represented section of the original polyline. That is, we measure the longest distance from a point along the new geometry to the original polyline. This is equivalent to computing the minimum epsilon-band around the represented section that contains the new edges. The total error induced by an offset of the harmony line is the maximum error over all rungs.

We sample offsets at regular distances, limiting the maximal offset in either direction to the extremal offset where at least one new vertex lies inside the span of the respective rung. We select the offset of the harmony line that minimizes the induced error.

**Maintaining a matching.** To compute the directed Hausdorff distance to the original polyline, we need a matching of the current geometry and the original polyline. Initially, this matching is the identity. We update the matching iteratively during the algorithm per rung

of the collapsed ladder. Let  $\langle s, t, u, v \rangle$  denote the four vertices associated with rung  $tu$ ; let  $\mu$  denote the current matching. Moreover, let  $\max \mu(w)$  be the last vertex on the matched geometry that is matched to  $w$ . Similarly  $\min \mu(w)$  is the first vertex matched to  $w$ . As  $tu$  is replaced by a new point  $p$ , the main concern is to re-assign the vertices in  $\mu(t)$  and  $\mu(u)$  to  $s$ ,  $p$ , or  $v$  such that the entire matching remains monotone. We compute the ELCFM between the new geometry  $\langle s, p, v \rangle$  and the original polyline from  $\max \mu(s)$  to  $\min \mu(v)$ . This “local” ELCFM is used to update the matching  $\mu$ . We use the Euclidean variant as the geometries may intersect and hence geodesic distances are unsuitable.

### 2.3 Iterative reduction: overall algorithm

Our algorithm iteratively collapses slope ladders as detailed above. We now describe the remaining high-level algorithmic details: selecting the slope ladder to collapse, criteria for stopping, and how to efficiently avoid intersections.

**Selecting a collapse.** We greedily select the collapse that causes the least visual change as measured by the (average) symmetric difference (or areal displacement) between the current geometry and the geometry after collapse; this is similar to existing iterative approaches, e.g. [6, 19]. As ladders with higher complexity naturally incur more (total) symmetric difference but also reduce the complexity (number of vertices/edges) more, we divide the total symmetric difference by the number of rungs  $c$ , to obtain the average symmetric difference per rung.

Note that our method for selecting the collapse geometry ensures that the result remains close to the input geometry, while our selection strategy ensures little visual change with respect to the current geometry. The advantage of the latter as a high-level strategy is that the sequence of simplifications changes as little as possible from step to step, thus improving the “stability” of the results. Simplifications of almost the same complexity will have similar layouts. Notably this is not guaranteed by the Hausdorff distance in the collapse computation as it compares only to the original polylines.

**Stopping criteria.** There are various options that may be used as a stopping criterion for the algorithm. We could (1) perform operations until the collapse cost exceeds a certain threshold, (2) disallow collapses that cause too large a Hausdorff distance, or (3) stop when the total number of vertices is at most a given parameter  $z \geq 2k$ . Following the techniques of [7], we observe that we can easily keep track of the performed operations in a data structure to efficiently reconstruct all intermediate results.

**Avoiding intersections.** To avoid introducing intersections between isolines or within the same isoline (R8), we perform only collapses that do not create intersections. We follow the algorithm described by [6] and maintain “blocking numbers”: this number stores the number of edges that are intersected by the new geometry if the collapse was performed. Thus, we perform only collapses with blocking number zero. The main idea is that, after initializing these, collapses change only few edges and only these changing edges need to be considered to update the blocking numbers. This is more efficient than testing intersections from scratch.

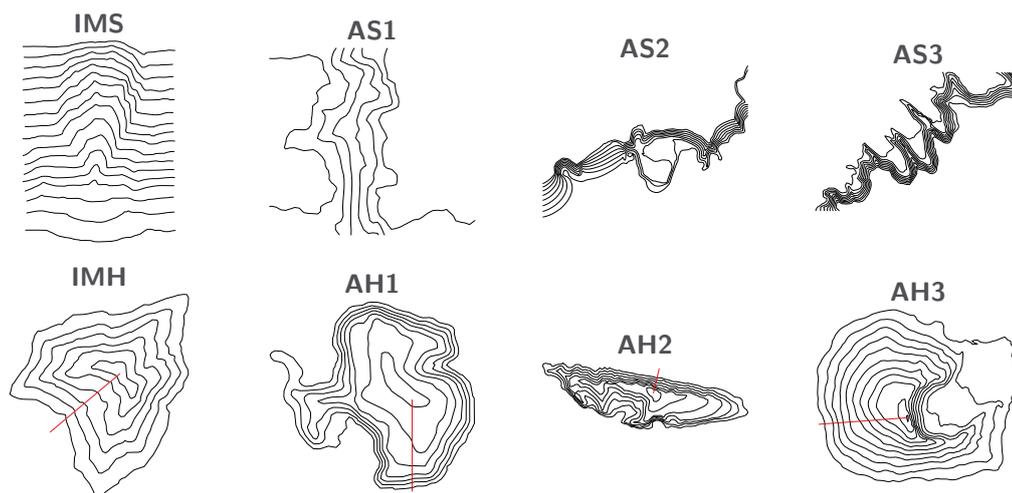
**Running-time analysis.** We assume that there are  $k$  polylines of  $n_h$  vertices each, with  $n = \sum_{h=1}^k n_h$  the total number of vertices. Our algorithm runs in  $O(n^2 \log n)$  time, using  $O(n)$  space. We first compute slope ladders with the GLCFM in  $O(n^2 \log n)$  time using the data structure of Guibas and Hershberger [14] and the algorithm by Buchin et al. [5].

We then initialize all collapses in  $O(n^2)$  using the approach of Alt et al. [1] to compute the Hausdorff distance. Finally, we iteratively collapse the best slope ladder and update any affected ladders in total  $O(n^2 \log n)$  time.

### 3 Results and discussion

We implemented our algorithm<sup>1</sup> and tested it with various isolines. We first qualitatively compare the results on different sets of isolines through a visual exploration of the properties. We then quantitatively measure how well harmony is maintained using our method of collapsing slope ladders compared to simplifying each isoline separately.

We showcase and measure results for eight different inputs. The first are two digitized textbook examples, namely two sets of isolines used by Imhof to illustrate isoline generalization practices [18, page 155]; we denote these by **IMS** (slope) and **IMH** (hilltop). **AH1** is a digitized set of contour lines for the whole of Antarctica at 500m contour intervals. The other five are snippets from the DEM of Antarctica [17], deriving isolines at a 150m contour interval using QGIS v.3.16; we denote these snippets by **AS1** through **AS3** (slopes) and **AH2** through **AH3** (hilltops). All inputs are illustrated in Fig. 7. The algorithm discussed considers polygonal lines, hence for our experiments we open any closed isoline. Though out of scope, the concepts can readily be extended to closed isolines and in future work this restriction will be lifted.

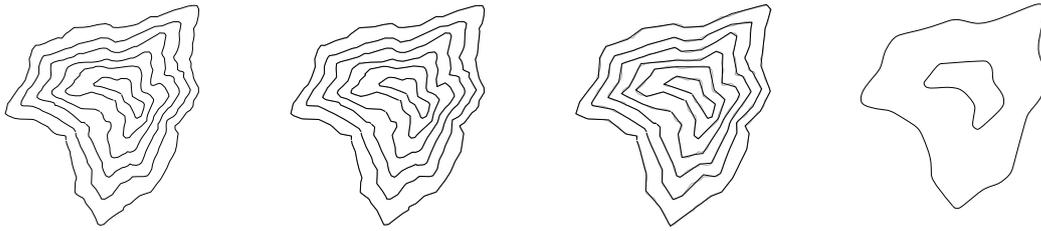


■ **Figure 7** Inputs used in this paper: four slopes (top) and four hilltops (bottom). Complexities range from 442 (**AS1**) to 2071 (**AS3**). Red lines indicate where hilltops were opened into polylines.

#### 3.1 Qualitative discussion

**Examples by Imhof.** We start our visual investigation with **IMH**; see Fig. 8. We directly observe that the effect of harmonization is minor when the detail-level is high. Due to the large number of vertices in the simplification, the harmony is readily present as there are sufficient vertices to always have nearby vertices on the adjacent isoline. Isoline detail is the primary factor. However, when we reduce the complexity further, the underlying patterns in the isolines become emphasized through harmony. As vertices are placed at similar locations

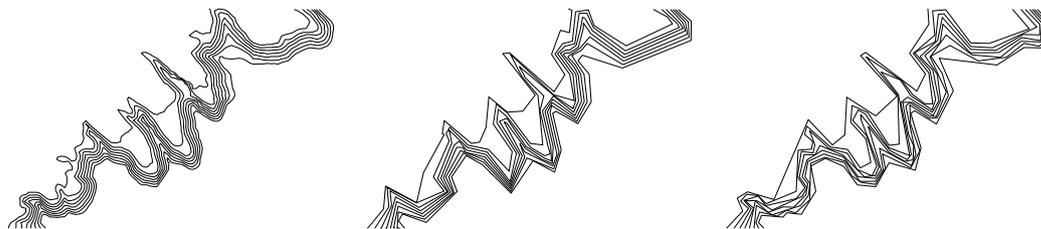
<sup>1</sup> <https://github.com/tue-alga/harmonious-simplification> (v0.1)



■ **Figure 8** (left) IMH with 664 vertices. (middle left) Simplification at 348 vertices does not visually benefit much from harmony. (middle right) Simplification at 150 vertices, with harmony emphasizing the underlying patterns. (right) Example simplification as given by Imhof (from [18]).

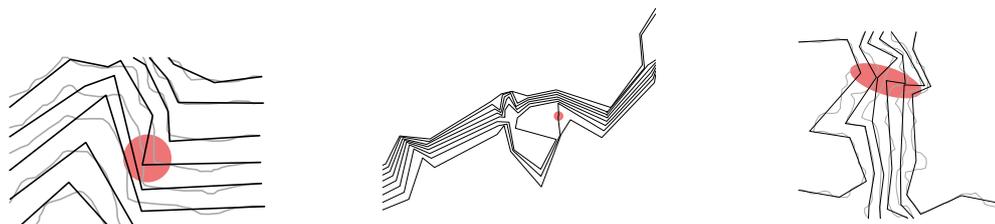
whenever isolines trace similar boundaries, higher-level structures visually start to form a coherent whole strengthening the relationship between adjacent isolines. The effect is visible for both IMH and IMS (see also Fig. 13) giving credit to the suitability of the approach.

**Visual complexity.** When considering more tight-packed slope lines, the benefits of harmony are even more apparent. In Fig. 9 we compare two simplifications of AS3 with approximately the same number of vertices. One of the simplifications is computed with our algorithm, the other uses the same simplification steps but collapses edges independently during the process. We observe that enforcing harmony captures the strong similarity between the isolines. Consequently, similar sections are also highly similar in the simplification in contrast to the independently simplified isolines. This coherence (1) strongly reinforces the underlying patterns, (2) leads to a lower visual complexity as isolines can implicitly be grouped, and (3) emphasizes the object represented by the group of isolines in contrast to the isolines themselves. The lack of harmony in the alternative result hinders the visual perception of the isolines as representing a single object as the visual clutter hides underlying patterns.



■ **Figure 9** (left) AS3 with 2071 vertices. (middle) Simplification with 246. (right) Simplification with 247 edges using our approach but collapsing a single edge at a time.

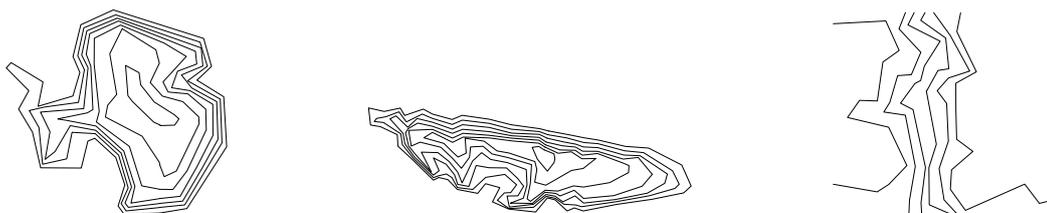
**Limitations.** Our algorithm sometimes makes minor mistakes that are undesirable (Fig. 10). Particularly, the individual simplification may be suboptimal compared to a potential individual simplification as the error margin could trivially be lowered by shifting single vertices (Fig. 10(a/c)), or insignificant edges are maintained even though they do not contribute to the overall simplification (Fig. 10(b)). We believe these may most likely be traced back to the following issues. By strictly enforcing area-preservation it is possible that a vertex is placed to compensate for a region being cut off through simplification (a). The problem can be exaggerated due to the stringent requirement that all newly placed vertices



■ **Figure 10** (left) area-preservation requirements may lead to poor vertex placement (**IMS** - 98 vertices), (middle) minor edges despite a high level of simplification (**AS2** - 123 vertices), (right) forcing vertices to be on a single-line can cause unnecessary errors (**AS1** - 50 vertices).

need to be on a single line (c). This may also affect the (im)possibility to simplify certain sections in union without a high error margin for one of the isolines, potentially leading to unnecessary minor edges (b).

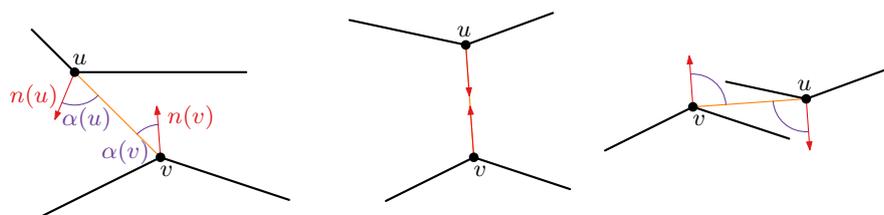
In future work, we plan to investigate more flexible treatment of slope ladders to overcome these issues. Overall we believe the method shows great potential as witnessed by its ability to suitably simplify many different isoline structures (see Fig. 11).



■ **Figure 11** Harmonious simplifications of (left) **AH1** with 150 vertices, (middle) **AS2** with 250 vertices, (right) **AS1** with 70 vertices.

### 3.2 Quantitative analysis

The aim of our algorithm is to maintain harmony as a visual cue that the isolines describe a single underlying object. In this section we check whether our algorithm is also measurably successful at maintaining harmony. We compare the results of our algorithm to a modified version in which each edge is treated “independently”. We perform the exact same collapsing procedure, but do so for only one edge at a time. We refer to this modified version as the *isolated algorithm*, contrasting our proposed *harmonious algorithm*.



■ **Figure 12** (left) The harmony measure for a matched pair  $(u, v)$  is the sum of the angles indicated in purple  $(\alpha(u) + \alpha(v))$ . Note that the direction of the normal is irrelevant for the score. (middle) A “perfect” pair, scoring zero. (right) A “worst” pair, scoring 180 degrees.

**Measuring harmony.** To support a quantitative comparison, we must quantify the harmony of a set of isolines. We use a simple measure that captures one of the facets of harmony. Particularly, we compute the GLCFM between each subsequent pair of isolines and then measure the harmony for each matched pair. Intuitively, a pair of vertices  $(u, v)$  is matched harmoniously if the segment  $uv$  is locally orthogonal to the isolines. Thus, we estimate a normal  $n(u)$  at vertex  $u$  and a normal  $n(v)$  at  $v$ . We then measure the minimal angle  $\alpha(u)$  (respectively  $\alpha(v)$ ) between the line spanned by  $uv$  and  $n(u)$  (respectively  $n(v)$ ) (see Fig. 12). The sum of  $\alpha(u)$  and  $\alpha(v)$  is the measured harmony for this pair. This score ranges from zero (“perfect” harmony) to 180 degrees (“worst” harmony).

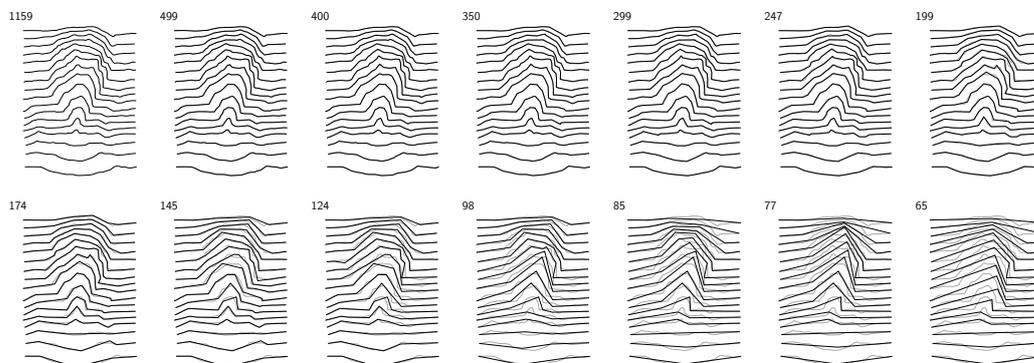
Our straight harmony line enables good harmony across multiple isolines: these vertices are expected to be matched in the eventual GLCFM and a vertex’s normal can be close to the matching lines to both its neighbors only if these lines have similar orientation.

**Measuring similarity.** Typical approaches to simplification quantify the quality of the resulting simplification by its (geometric) similarity to the input. We may hence generally use geometric similarity as a measure of quality; for our evaluation we use the *continuous* Fréchet distance between a computed simplified isoline and its original geometry for this purpose. We normalize this distance between instances using the diagonal of the bounding box of the input. We use the continuous version, as the strongly differing complexities between the two polygonal lines causes the discrete version to not measure shape similarity very well.

**Setup.** We track how harmony develops throughout both algorithms as complexity decreases. We first run our harmonious algorithm until it is stuck, saving intermediate results when the complexity first reaches or drops below one of a given sequence of desired complexities<sup>2</sup>. An example sequence obtained in this manner is shown in Fig. 13. Since a collapse may reduce the total complexity by the complexity of the ladder, the resulting maps do not necessarily have the exact desired complexities. We then run the isolated algorithm, using the same procedure but as desired complexities, those actually achieved by the harmonious algorithm.

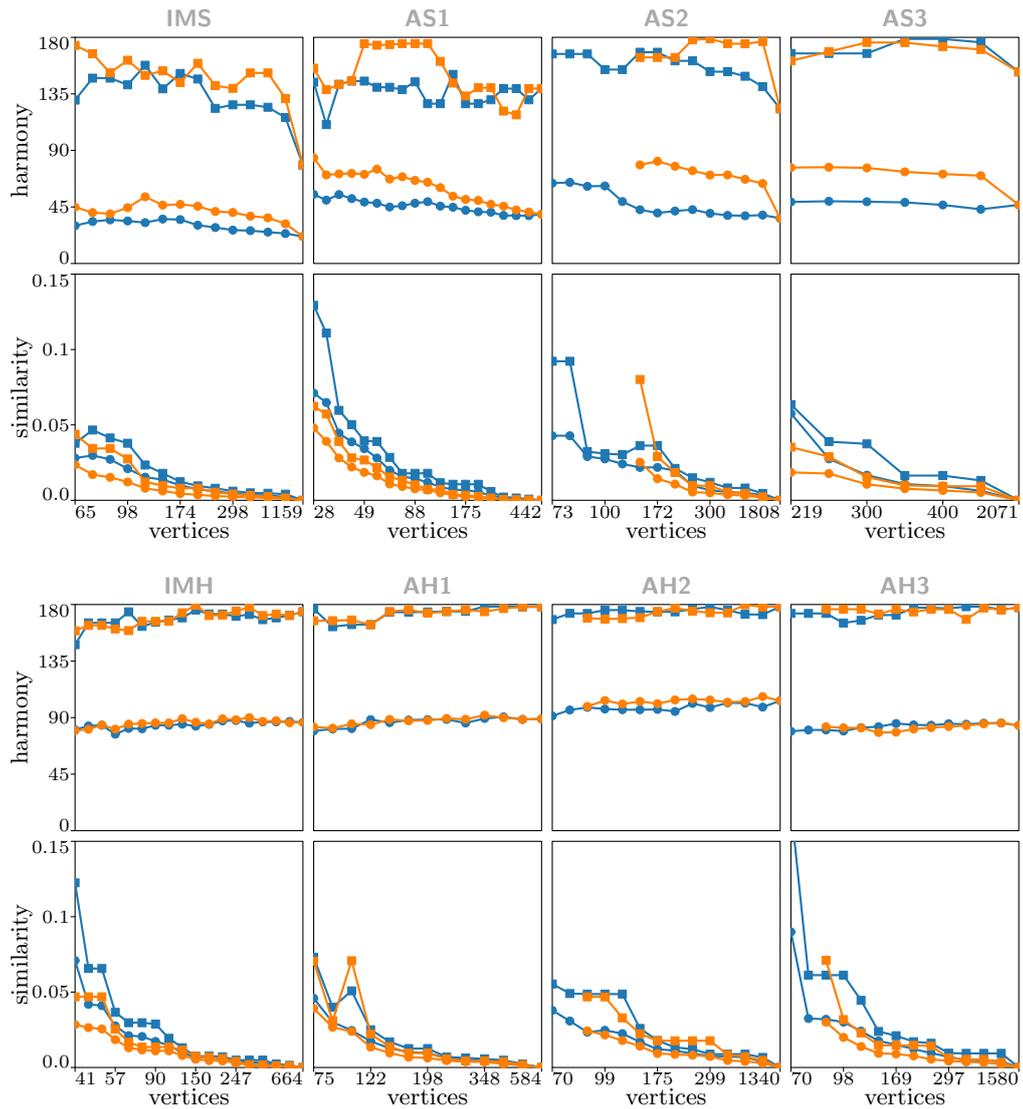
We measure the harmony and similarity as both algorithms progress, that is, in their sequence of maps. We plot the average and maximum harmony over all matched pairs and average and maximum similarity over all isolines as a function of complexity (Fig. 14).

<sup>2</sup> 500, 400, 350, 300, 250, 200, 175, 150, 125, 100, 90, 80, 70, 60, 50, 45, 40, 35, 30, 25, 20



■ **Figure 13** Results of IMS using our harmonious algorithm; numbers indicate complexity.

8:14 Harmonious Simplification of Isolines



■ **Figure 14** Average (circles) and maximum (squares) harmony and similarity as the algorithms simplify the input; lower values are better. Blue: harmonious algorithm; orange: isolated algorithm. Note that the horizontal axis is not linear. The maximum similarity (0.27) for the result of our harmonious algorithm on **AH3** with 81 vertices has been cropped.

**Observations.** First, we observe that the harmonious algorithm tends to perform better for harmony than the isolated method, but the similarity distance is typically slightly worse. This stands to reason as a collapse may sacrifice similarity to improve harmony. However, the effect on similarity is rather minor, until we reach the lower levels of complexity.

Second, the average and maximum Fréchet distance behave very similarly and tend to not differ much until the algorithm reaches the lowest complexities. This suggests that the geometric distortion our algorithm causes remains evenly spread between the isolines.

Third, we observe that for the isolated algorithm the average harmony tends to deteriorate. For the harmonious algorithm the harmony tends to stay roughly constant, though some of the slopes do show a bit of a deterioration as well (**AS1–AS2** mostly).

Fourth, we observe that the isolated algorithm often is stuck earlier, never reaching the number of vertices that the harmonious algorithm can reach; this is the case for **AS2–AS3**, **AH1–AS3**. This may seem somewhat counterintuitive: staying closer to the original geometry should generally help avoid intersections. There are two possible explanations here: (1) by collapsing entire slope ladders, we may still perform a collapse, even if collapsing any of the rungs in isolation would cause intersections; (2) by incorporating harmony, there is a better relation between the isolines and thus performing a collapse (even if in isolation) is less likely to cause intersections. We believe the first explanation is most likely the strongest effect here, but the second explanation may warrant further investigation in future work.

## 4 Conclusion and future work

We have presented an algorithm for simplifying isolines harmoniously, i.e., acknowledging that they describe a single object and that structures of this object should be reflected clearly in the result. Our evaluation shows the efficacy of our method, but also marks some current limitations to be overcome in future work. In addition to these, we observe that simplifying larger features (more than a single slope or hilltop) requires more investigation. For example, if a hill has two hilltops, we may need to match the isoline that encompasses both partially to one hilltop and partially to the other. In similar spirit, even between two isolines, large flat areas should possibly be omitted in the matching as maintaining harmony over a large distance is counter to the underlying principle; again, some form of partial matching may be useful here. As object identification in isoline systems is quite advanced and computationally efficient via contour trees and similar Reeb-graph-like structures, the starting and end-point problems for partial matchings within our method remain future work.

---

### References

- 1 Helmut Alt, Bernd Behrends, and Johannes Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13(3):251–265, 1995.
- 2 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.
- 3 Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959.
- 4 Prosenjit Bose, Sergio Cabello, Otfried Cheong, Joachim Gudmundsson, Marc J. van Kreveld, and Bettina Speckmann. Area-preserving approximations of polygonal paths. *Journal of Discrete Algorithms*, 4(4):554–566, 2006.
- 5 Kevin Buchin, Maike Buchin, Wouter Meulemans, and Bettina Speckmann. Locally correct Fréchet matchings. *Computational Geometry*, 76:1–18, 2019.
- 6 Kevin Buchin, Wouter Meulemans, André Van Renssen, and Bettina Speckmann. Area-preserving simplification and schematization of polygonal subdivisions. *ACM Transactions on Spatial Algorithms and Systems*, 2(1):Article No. 2, 1–36, 2016.
- 7 Thomas C. van Dijk, Arthur van Goethem, Jan-Henrik Haunert, Wouter Meulemans, and Bettina Speckmann. Map schematization with circular arcs. In *Proc. International Conference on Geographic Information Science*, pages 1–17, 2014.
- 8 Regina Estkowski and Joseph S. B. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *Proc. 17th Symposium on Computational Geometry*, pages 40–49, 2001.
- 9 Julien Gaffuri, Cécile Duchêne, and Anne Ruas. Object-field relationships modelling in an agent-based generalisation model. In *Proc. 12th Workshop on Generalisation and Multiple Representation*, 2008.
- 10 Aji Gao, Jingzhong Li, and Kai Chen. A morphing approach for continuous generalization of linear map features. *Plos one*, 15(12):e0243328, 2020.

- 11 Arthur van Goethem, Wouter Meulemans, Andreas Reimer, and Bettina Speckmann. Simplification with parallelism. In *Proc. 23rd ICA Workshop on Generalisation and Multiple Representation*, 2020.
- 12 Arthur van Goethem, Wouter Meulemans, Bettina Speckmann, and Jo Wood. Exploring curved schematization of territorial outlines. *IEEE Transactions on Visualization and Computer Graphics*, 21(8):889–902, 2015.
- 13 Alan H. Goldman. Aesthetic qualities and aesthetic value. *The Journal of Philosophy*, 87(1):23–37, 1990.
- 14 Leonidas J. Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989.
- 15 Eric Guilbert. Feature-driven generalization of isobaths on nautical charts: A multi-agent system approach. *Transactions in GIS*, 20(1):126–143, 2016.
- 16 Eric Guilbert, Julien Gaffuri, and Bernhard Jenny. Terrain generalisation. In *Abstracting geographic information in a data rich world*, LNCG, pages 227–258. Springer, 2014.
- 17 Ian M. Howat, Claire Porter, Benjamin E. Smith, Myoung-Jong Noh, and Paul Morin. The reference elevation model of Antarctica. *The Cryosphere*, 13(2):665–674, 2019.
- 18 Eduard Imhof. *Kartographische Geländedarstellung*. De Gruyter, 1965.
- 19 Barry J. Kronenfeld, Lawrence V. Stanislawski, Barbara P. Buttenfield, and Tyler Brockmeyer. Simplification of polylines by segment collapse: Minimizing areal displacement while preserving area. *International Journal of Cartography*, 6(1):22–46, 2020.
- 20 Zhilin Li and Haigang Sui. An integrated technique for automated generalization of contour maps. *The Cartographic Journal*, 37(1):29–37, 2000.
- 21 Maarten Löffler and Wouter Meulemans. Discretized approaches to schematization. In *Proc. 29th Canadian Conference on Computational Geometry*, pages 220–225, 2017.
- 22 Thomas Mendel. Area-preserving subdivision simplification with topology constraints: Exactly and in practice. In *Proce. 20th Workshop on Algorithm Engineering and Experiments*, pages 117–128, 2018.
- 23 Paulo Raposo. Scale-specific automated line simplification by vertex clustering on a hexagonal tessellation. *Cartography and Geographic Information Science*, 40(5):427–443, 2013.
- 24 Andreas Reimer. *Cartographic modelling for automated map generation*. PhD thesis, Technische Universiteit Eindhoven, 2015.
- 25 Timofey E. Samsonov. Automated conflation of digital elevation model with reference hydrographic lines. *ISPRS International Journal of Geo-Information*, 9(5), 2020.
- 26 Timofey E. Samsonov, Sergey Koshel, Dmitry Walther, and Bernhard Jenny. Automated placement of supplementary contour lines. *International Journal of Geographical Information Science*, 33(10):2072–2093, 2019.
- 27 Wilhelm Schüle. Zur Maßstabsfrage des neuen schweizerischen Kartenwerkes, mit einem Nachtrag und Anhang zur Kurvendarstellung auf topographischen Karten. In *Jahresbericht der Geographischen Gesellschaft von Bern, Bd. XXVIII*, pages 31–53, 1929.
- 28 Andriani Skopeliti, Lysandros Tsoulos, and Shachak Pe’eri. Depth contours and coastline generalization for harbour and approach nautical charts. *ISPRS International Journal of Geo-Information*, 10(4):197, 2021.
- 29 Xiaohua Tong, Yanmin Jin, Lingyun Li, and Tinghua Ai. Area-preservation simplification of polygonal boundaries by the use of the structured total least squares method with constraints. *Transactions in GIS*, 19(5):780–799, 2015.
- 30 Guillaume Touya, Hugo Boulze, Anouk Schleich, and Hervé Quinquenel. Contour lines generation in karstic plateaus for topographic maps. In *Proc. International Cartographic Conference*, volume 2, pages 1–8, 2019.
- 31 Dražen Tutić and Miljenko Lapaine. Area preserving cartographic line generalization. *Kartografija i geoinformacije (Cartography and Geoinformation)*, 8(11):84–100, 2009.
- 32 Dražen Tutić, Matjaž Štanfel, and Tomislav Jogun. Automation of cartographic generalisation of contour lines. In *Proc. 10th ICA Mountain Cartography Workshop*, pages 65–77, 2017.

# Navigating Your Way! Increasing the Freedom of Choice During Wayfinding

Bartosz Mazurkiewicz ✉ 

Geoinformation, TU Wien, Austria

Markus Kattenbeck ✉ 

Geoinformation, TU Wien, Austria

Ioannis Giannopoulos ✉ 

Geoinformation, TU Wien, Austria

---

## Abstract

Using navigation assistance systems has become widespread and scholars have tried to mitigate potentially adverse effects on spatial cognition these systems may have due to the division of attention they require. In order to nudge the user to engage more with the environment, we propose a novel navigation paradigm called *Free Choice Navigation* balancing the number of free choices, route length and number of instructions given. We test the viability of this approach by means of an agent-based simulation for three different cities. Environmental spatial abilities and spatial confidence are the two most important modeled features of our agents. Our results are very promising: Agents could decide freely at more than 50% of all junctions. More than 90% of the agents reached their destination within an average distance of about 125% shortest path length.

**2012 ACM Subject Classification** Information systems → Decision support systems; Computing methodologies → Agent / discrete models; Information systems → Location based services

**Keywords and phrases** Agent-based Simulation, Wayfinding, Free Choice Navigation

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.9

## 1 Introduction

Adaptive route instructions for pedestrian wayfinders have seen considerable research interest [7]. At the same time, empirical evidence has been collected which suggests an adverse impact of wayfinding assistance systems on spatial cognition and knowledge acquisition (see e.g., [6]). Routing paradigms which allow wayfinders to make an increased number of decisions and, therefore, reduce the number of instructions given as much as reasonable would be one option to remedy this effect. We explore one possible solution to this problem and propose a pedestrian navigation paradigm which balances the number of given instructions and the freedom of choice left to the user at junctions. We explore the feasibility of this paradigm by means of an agent-based simulation. At the start, an agent is provided with a destination vector, similar to someone pointing to the destination when asked for instructions. Once an agent started walking they do not need to stick to a predefined route, hence they will not suffer from *on-route uncertainty* [22] but are free to make their own decisions at junctions. However, a route instruction will be provided if odds are increased that an agent, based on its current state, is likely to choose a not reasonably good branch at a given junction. The presented paradigm aims for less instructions and more free choices along the route – regardless the way instructions are phrased or the modality they are given in.

Based on a comparison of our routing paradigm (labelled *free choice navigation – FCN*) to the turn-by-turn (TBT) technique, which is prevalent in commercial wayfinding assistance systems, our contribution is threefold:



© Bartosz Mazurkiewicz, Markus Kattenbeck, and Ioannis Giannopoulos;  
licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 9; pp. 9:1–9:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1. We present a new navigation paradigm (*free choice navigation*);
2. We provide evidence that agents reach their destinations within a reasonable departure from the length of the shortest path using our approach and
3. We quantify the number of instructions we save/add compared to the baseline (TBT).

Successfully testing of our approach has a potentially important implication: Reducing the number of instructions would provide users of wayfinding systems with more time to observe the environment while, simultaneously, forcing them to engage more with it as they have to make their own decisions. Based on prior evidence (see e.g., [33]), this is expected to result in increased spatial knowledge acquisition (see section 7 below). It is important to note, though, that our approach is meant to be used in a leisure scenario, i.e. in situation in which wayfinders do not feel any time pressure.

## 2 Related Work

Given our research goal, we discuss three different branches of related work. First, literature on reducing the number of TBT navigation instructions or adapting their structure/presentation is reviewed. Second, we discuss literature using the beeline/as-the-crow-flies navigation approach. Lastly, we review agent-based simulations modeling pedestrian behavior. Taken together, this review reveals important factors which need to be considered when modeling agents and its accompanying mechanics (see also Section 4).

### 2.1 Enhancing TBT Instructions

In recent years, evidence has been collected which suggests that the use of navigation systems may have an adverse impact on spatial cognition and orientation (see e.g., [6, 21]). This effect is commonly explained by the need to divide the attention between the navigation system and the environment when following a pre-determined route (see e.g., [13]). Possible remedies regarding TBT navigation systems are enhancing instructions with additional information (see e.g., [16, 36]), reducing the number of instructions [28], combining them with different interaction techniques (see e.g., pointing [25]) or providing haptic or audio feedback (see e.g., [12, 14]). All these approaches assume a predefined route. This is in contrast to our paradigm according to which users can make their own spatial decisions to a large extent.

### 2.2 Beeline Navigation

There are alternatives to TBT navigation approaches, e.g., using the beeline to the destination. One particularly important idea is the so-called least-angle strategy, which was for the first time thoroughly studied by Hochmair and Frank [20] using a simulation study. According to this strategy a user chooses the option with the least angle with respect to the (believed) destination vector. The least-angle strategy has been studied with respect to various implementations: [29] and [8] report on prototype navigation systems for pedestrians which use vibro-tactile feedback devices to indicate the beeline. Either system guides users successfully to their destination. Both systems allow for free exploration but none of them control for an upper path length limit. This is in contrast to our approach: We try to find a compromise between free exploration and maximum path length while determining the point in time at which an instruction needs to be given. Savino and colleagues [30] compare TBT and two different implementations of the beeline approach for cyclists, one of which provides visual cues when the beeline differs from the shortest path branch. The latter approach enhances user confidence. The beeline approach was preferred for leisure scenario. This is in line with

prior evidence suggesting that it is important for pedestrians to optimize both, distance and angle [31] and that the environmental complexity [15] has also a major impact on route choice behavior. Our approach takes these findings into account (see Section 4) by providing the beeline at the starting point and providing further wayfinding assistance as needed.

### 2.3 Simulation Studies and Models of Pedestrian Behavior

As mentioned above, the beeline approach was studied by Hochmair and Frank [20] using a simulation study regarding perception errors. Based on this work, Hochmair [19] analyzed the effectiveness of the least-angle strategy with a simulation study for different transportation modes. The results suggest a limited usability of this approach in real life (human perception and memory errors). Our simulation takes these findings into account by modeling human errors and actively supporting the user if help is needed, moreover, it incorporates environmental spatial abilities and spatial confidence (see Section 4). Other agent-based wayfinding studies frequently focus on collision avoidance (see e.g., [35]), on evacuation (see e.g., [37]) or on the interplay between navigational instructions, the environment and the agent (see e.g., [34, 22]), including research on route choice behavior based on different levels of cognitive maps [10]. Generally speaking, the findings of these papers are based on the assumption that a path a user should follow exists. Again, this is in contrast to our approach. Neither do agents receive an instruction at every decision point, nor is there a predefined path. Our agents are also not equipped with prior knowledge of the environment nor do they know the true destination direction at every junction (which is in contrast to, e.g., the models used by Kneidl [23]), i.e., they are considered to be unfamiliar and, hence, have no cognitive map.

## 3 Simulation Preparation and Baseline Condition

In order to test<sup>1</sup> our hypotheses (see Section 5) we run an agent-based simulation (non multi-agent) with two conditions. The main focus lies on whether agents do reach their destination with our approach. Consequently, we will not model effects on spatial cognition, e.g. spatial knowledge acquisition as this aspect will be scrutinized in a real-world study (see Section 7). The simulation study was run in three cities with three different network types [32], namely Djibouti City (Djibouti, type: *irregular*), Vienna (Austria, type: *high transit*) and Mexico City (Mexico, type: *checkerboard*). For each city 100 random routes with a length ranging between 500m and 5000m were chosen in order to test the approach on shorter and longer routes. We tested two conditions (*TBT* vs. *free choice navigation*): Each condition was tested with 3000 agents, i.e., there are 6000 different agents (between subject design). The agents are constructed based on the the so-called BDI-framework [27] for practical reasoning: The agents simulate pedestrians who have **B**eliefs about where their destination is located; agents **D**esire to reach their destination; and, finally, they act based on **I**ntentions, i.e., at each intersection they reason about which decision they need to take in order to reach their goal based on their beliefs.

### 3.1 Software and Data

Several agent-based modeling and simulation frameworks exist [1]. For our approach we utilized python 3.8.8 using the package networkx (v2.5.) [17], avoiding frameworks providing graphical interfaces since this was not relevant for the presented study. The raw network

---

<sup>1</sup> The terms *junction* and *intersection* are used interchangeably, as well as *option* and *branch* are.

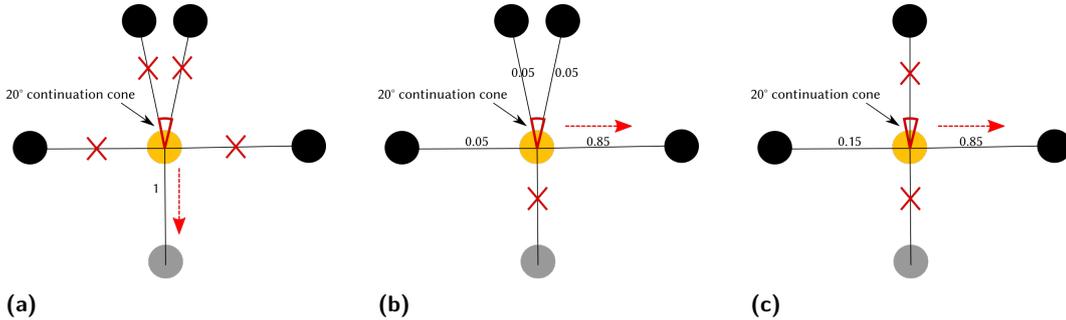
data were downloaded from OpenStreetMap (OSM). The intersections were calculated with the Intersections Framework [11], whereas street segments were extracted with a custom script. With those components each city could be represented as a networkx graph.

### 3.2 Following Navigation Instructions

In this subsection the modeling aspect which is shared across both simulation conditions, is explained: In either condition an agent has the ability to follow navigation instructions. This ability ( $nav\_instr$ ) is modeled using a value in the range  $[0.8; 1.0]$  and represents the probability of taking the branch indicated in the route instruction. This ability is expected to be high because navigation instructions are followed on a daily basis by millions of users, hence the interval between  $[0.8, 1.0]$ . We use a uniform distribution in case of the baseline, whereas a normal distribution is assumed in case of our *free choice navigation* approach (see Section 4).

### 3.3 Baseline

The baseline our approach is compared with is a TBT navigation system, i.e., agents receive navigation instructions only at turning points. Agents in this condition have a single attribute called  $nav\_instr$  which represents their ability to follow a route instruction correctly. It is represented as a value within  $[0.8; 1.0]$  drawn from a uniform distribution, because no figures were found on how well people can follow turn-by-turn navigation instructions in general (i.e., a mean value based on empirical grounds is not available).



■ **Figure 1** The decision mechanism for the baseline condition. current node: yellow (in the center); previous node: grey; given instruction: red arrow; excluded branches: red cross; annotations represent the probability of a branch to be taken: (a) turn-around instruction (always performed correctly); (b) Sample turn instr. 1 with two straight ahead options; (c) Sample turn instr. 2 with only one straight ahead option.

There are three options for an agent at any junction: **(1) no instruction** means continue straight ahead (within a  $20^\circ$  cone); **(2) turn-around instruction** which are always followed correctly (see Fig. 1a); **(3) turn instruction** which is interpreted as a weighted random choice (see Fig. 1b and 1c). The decision which segment is taken is modeled as a multi-step process, which starts only if a turn instruction is given: First, all potential options to follow are identified. This means, the option the agent is coming from is excluded because it would have been otherwise a turn-around instruction. If there is only one straight ahead option, it is also excluded too (see Fig. 1c) because no instruction would have been given if continuing straight would have been the correct option. Next, probabilities are assigned to each option: The agent's ability to follow navigation instructions ( $nav\_instr$ ) is assigned as

a probability to the option which is indicated as correct by the turn-instruction. All other  $n$  remaining options have a probability of  $\frac{1-nav\_instr}{n}$ . Finally, given these probabilities a weighted random choice is performed and the agent moves to the next junction and the procedure is repeated until the agent reaches its destination.

## 4 Free Choice Navigation Approach

In this section, a detailed account of the *free choice navigation* approach will be given. We, first, describe the properties of our agents and then, we move on to explain the decision mechanism of our agents in detail. In order to mitigate potentially adverse effects on spatial cognition by the usage of wayfinding assistance systems (see, e.g., [21]) we propose the *free choice navigation* approach. This paradigm nudges users to engage with the environment by balancing the number of free route choices against a given maximum distance threshold.

### 4.1 Agent Properties

Due to the nature of our approach, modeling the agents is more complex than in the baseline condition. Agents have five different properties, each of which is detailed below:

**Belief Vector (*belief\_vec*)** represents a subjective vector from the current junction to the believed location of the destination which, in turn, depends on the agent's current orientation. In contrast, the true destination vector (*true\_dest\_vec*) is the beeline from the current junction to the true destination location. These two vectors usually differ (see [20] for this claim and Figure 3 for an example).

**Environmental Spatial Abilities (*env\_sp\_ab*)** follow a normal distribution ( $M = 0.5$ ,  $SD = 0.2$ ; co-domain  $[0; 1]$ ). They have an impact on the belief vector. They are needed to “form a coherent mental representation” of the environment [39]. To the best of our knowledge, there is no evidence about the distribution of Environmental Spatial Abilities, therefore we assumed a normal distribution.

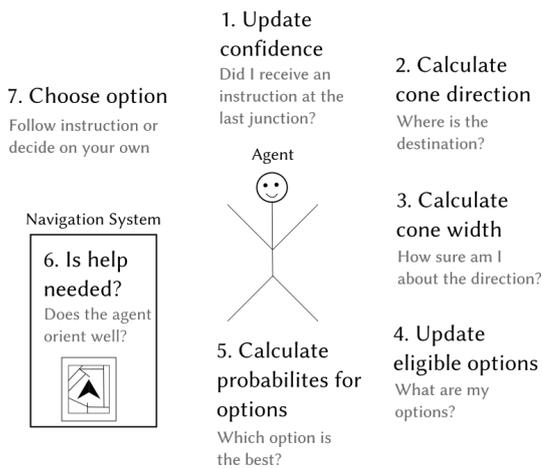
**Spatial Confidence (*conf*)** refers to an agent's confidence [39] about the destination direction ( $[0.0; 1.0]$ ). Prior evidence indicates (see, e.g., [26]) that self-reported tests on spatial abilities have been a good performance predictor. We, therefore, model an agent's spatial confidence based on its environmental spatial abilities: It seems to be plausible to assume that agents which are good wayfinders would indeed have a high self-confidence in knowing the destination direction. Therefore, the maximum (*max\_conf*) and minimum (*min\_conf*) confidence level of an agent are related to *env\_sp\_ab* and are set to  $env\_sp\_ab \pm 0.2$  (co-domain  $[0; 1]$ ).

**Ability to Follow Navigation Instructions (*nav\_instr*)** represents the ability to follow a navigation instruction (see [34]; see also baseline agents). For this condition it is equal to  $env\_sp\_ab * 0.2 + 0.8$ ; in order to have the same range as for the baseline condition.

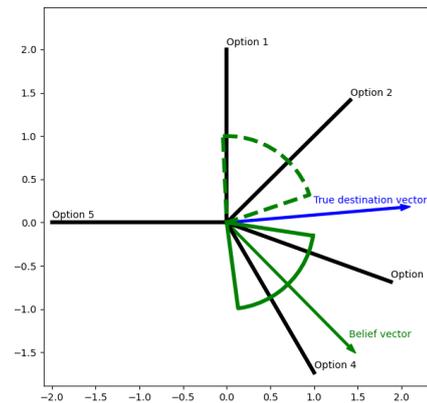
**Memorized Dead Ends** Agents are able to remember dead ends they have already taken in order to avoid back and forth movement (see [19]).

### 4.2 Initializing Agents

Before an agent starts a trial, both, its environmental spatial abilities (*env\_sp\_ab*) and its initial spatial confidence (*conf*) value are randomly set according to the intervals described above. Additionally, information about the destination direction is obtained, i.e., *belief\_vec* and *true\_dest\_vec* are pointing to the same direction. In each city the same population of agents is used. A trial is considered successful if and only if an agent has reached the destination deviating no more than a predefined threshold from the shortest path length.



■ **Figure 2** Overview of the decision mechanism for the *free choice navigation* condition.



■ **Figure 3** The agent will consider options 3/4 (branches intersect green solid line polygon). The dashed polygon represents the alternative cone (see step 2).

### 4.3 Decision Mechanism

The key geometric object we use to implement the reasoning of an agent at each intersection is a *cone*. This polygon represents where the agent believes the destination lies in (see Figure 3). The width of the cone (*cone\_width*) represents how certain an agent is about *belief\_vec*. It is, hence, influenced by *nav\_instr*, *conf* and environmental complexity (*env\_comp*, see below) and ranges between 360°(total confusion) and 1°(very sure).

Figure 2 provides an overview of the reasoning steps an agent and the simulated navigation system (only step 6) execute. It is important to note that this process will start if and only if the current node is not the goal node nor the current traversed path length is equal or greater than the maximum allowed path length. In both cases the trial would end immediately.

Having said this, the reasoning process starts with the update of the spatial confidence (step 1), based on whether an instruction was issued at the previous junction. Step 2 focuses on the update of its assumption about where the destination is located at. In Step 3 the certainty about this belief vector is modeled. Steps 4 deals with determining the eligible options, whereas a probability is assigned to each of these branches in step 5. Step 6 represents the most important part of this reasoning process, as is now determined whether an instruction should be issued. In a final step, a branch is chosen by the agent (step 7).

The detailed behavior can be described as follows:

#### Step 1: Update Spatial Confidence

The spatial confidence (*conf*) updates at every intersection<sup>2</sup>. The amount in change is calculated according to equation 1 (as mentioned above, its co-domain is restricted to  $[env\_sp\_ab \pm 0.2]$ ). If an agent received an instruction at the last intersection, the spatial confidence decreases, otherwise it increases. The rationale behind this mechanism is that wayfinders may perceive the fact that they received a route instruction as sign that they have taken wrong decisions in the past. This is, of course, intertwined with spatial abilities and, hence, agents with higher spatial abilities are less prone to losing their spatial confidence.

$$conf\_corr\_term = 0.05 * (1 - env\_sp\_ab) + 0.01 \quad (1)$$

<sup>2</sup> The spatial confidence will not be updated for the first and the second intersection as no instruction is given at these junctions, see step 6.

### Step 2: Calculate Cone Direction

The cone bisector is calculated according to the belief vector (*belief\_vec*, see eq. 2). In line with prior evidence [18], the agent's orientation deteriorates with distance (the number of traversed junctions) but less so for agents with higher *env\_sp\_ab*. For each intersection, this belief vector is rotated away from the *true\_dest\_vec* by random choice. The rationale for this randomness is that there is no apriori knowledge about in which direction the erroneous *belief\_vec* is rotated (see Figure 3).

$$belief\_vec = true\_dest\_vec \pm \left(1 - \frac{1}{\#traversed\_junctions}\right) * (1 - env\_sp\_ab) * 180^\circ \quad (2)$$

### Step 3: Calculate Cone Width

The width of the cone (*cone\_width*) represents how confident an agent is regarding the direction to the destination. It is influenced (see eq. 3) by the agent's spatial confidence and the environmental complexity of the current junction (*env\_comp*) whose impact is moderated by the agent's spatial abilities (the lower the environmental spatial abilities the higher the impact environmental complexity shows on the cone width). Before the whole simulation starts each junction in a city will be assigned a normally distributed environmental complexity (*env\_comp*,  $M = 0.5$ ,  $SD = 0.2$ ) which will remain unchanged during all trials. A node for which *env\_comp* = 0.5 is, therefore, considered as decision point having an average environmental complexity. Based on the *cone\_width* and the *belief\_vec* the actual cone geometry is calculated (see Figure 3).

$$cone\_width = 360^\circ - (conf * 360^\circ) + (env\_comp - 0.5) * 360^\circ * (1 - env\_sp\_ab) \quad (3)$$

### Update Eligible Options (Step 4) and Calculate Probabilities for Eligible Options (Step 5)

In order to find the set of eligible branches, agents exclude any already visited dead end (memorized options) from the decision process (Step 4). Having done so, the cone is checked for eligible options and angles with respect to the *belief\_vec* are calculated for each branch (Step 5). Three different cases are distinguished:

**Option 1: No branch in cone** All eligible options are taken into account and the angles between them and the *belief\_vec* are calculated (least-angle strategy [20]).

**Option 2: Exactly one branch in cone** This branch is assigned a probability of 1.0.

**Option 3: More than one branch in cone** As in this case an agent can choose from several branches inside the cone, angles are found for these by analogy with option 1.

For options 1 and 3, these angles, denoted as *opt\_ang*, need to be converted to probabilities using three further steps.

1. All angles are inverted with respect to the maximum angle within the cone which is given by  $\frac{cone\_width}{2}$ , thereby favoring angles which are closer to *belief\_vec*:  $inv\_angles = \frac{cone\_width}{2} - opt\_ang$ . Subsequently, each inverted angle will be normalized (division by the sum of all inverted angles in the list) in order to ensure that the sum of all probabilities equals one.
2. In order to avoid back and forth movement along the same branch (see [19]), a factor called *already visited penalty factor* (*vis\_pen*) is applied to the probability of any edge which has already been visited. The lower the value the higher the penalty. Again, the probabilities are normalized.

3. A final selection step for branches is needed as several branches may show almost the same angle to the bisector of the cone geometry, i.e., whose angles can hardly be distinguished by humans. Therefore, only those branches with a probability greater than  $max\_probability - 0.1$  are selected and then the remaining probabilities are normalized.

The above procedure leads to the final set of branches  $FS$  with their final probabilities assigned. These will subsequently be used to decide if an agent needs an instruction or not.

### Step 6: Determine Whether an Instruction is Given

Given the probabilities and several additional parameters a decision is made whether an instruction is given to the agent<sup>3</sup>. An instruction is provided for any of the following reasons, i.e., the list of reasons is checked one-by-one in the order given below:

1. The shortest path from the current junction to the destination leads over a single segment, i.e., the instruction is given in order to support destination recognition (see [9]).
2. Let  $sp\_length$  denote the shortest path from the start to the destination; let further  $dist\_walked$  denote the distance an agent has walked from the start to the current junction. Finally, let  $max\_dist$  denote the constant factor which determines the allowed deviation from  $sp\_length$  still rendering a trial successful. Then, the *current buffer* ( $curr\_buffer$ ), i.e., the distance an agent can walk from the current junction to the destination in order to still successfully finish a trial is given by equation 4.

$$curr\_buffer = max\_dist * sp\_length - dist\_walked \quad (4)$$

Let, furthermore,  $buff\_fact$  (co-domain:  $[0; 1]$ ) denote a factor which is used to balance the number of given instructions against the number of agents that will arrive (given the presupposition that the probability of a successful trial increases for agents who receive more instructions). This means, this factor is used to account for the risk that the agent will exceed the available buffer: An instruction will be given if the current buffer multiplied by the buffer factor is lower than the length of the shortest path from the current junction (see eq. 5).

$$curr\_buffer * buff\_fact < sp\_length\_curr\_jct \quad (5)$$

3. For the same reason other branches are examined (step 2 is introduced in order to save computation time). We evaluate this case based on two steps. We, first, create the set of acceptable branches  $AB$  which includes all branches which fulfill inequality 7 in which  $sp\_over\_i$  denotes the sum of the length of the shortest path between the upcoming intersection  $i$  and the destination and the length of the edge from the current junction to  $i$  (see eq. 6).

$$sp\_over\_i = sp\_length_i + len(edge\_to\_i) \quad (6)$$

$$sp\_over\_i \leq curr\_buffer * buff\_fact \quad (7)$$

Subsequently, we sum the probabilities of all branches  $b \in AB$ . If this sum is smaller than a predefined threshold called  $pos\_sum$ , which is kept constant across trials, then an instruction is given.

---

<sup>3</sup> There is no instruction given at the first two nodes the agent traverses as it is reasonable to assume that the direction of the destination is still evident.

4. An instruction is, furthermore, given if an edge exists in the *final selection*  $FS$  (see step 5 above) for which the shortest path from the end node of this edge leads over the current node. Let the current junction be denoted as  $j_{i-1}$ ; let the upcoming junction of a branch  $b_i$  which is part of the final selection set be denoted as  $j_i$ . An instruction will be given if the shortest path from  $j_i$  would lead over  $j_{i-1}$  because this will save agents a loss of *curr\_buffer* due to avoidable detours.
5. Given the fact that the number of agents with a successful trial must be maximized, an instruction is also provided if agents have a high probability of choosing a branch which will be costly in terms of the consumed buffer while there would have been “cheaper” alternatives. Let *rem\_len\_buf* (Remaining Length Buffer, RLB) denote the difference (see eq. 8) between the *curr\_buffer* and the *sp\_over\_i* (see eq. 6).

$$rem\_len\_buf = curr\_buffer - sp\_over\_i \quad (8)$$

**Step 1:** For all branches  $b \in FS$  the ratio between the RLB of the branch with the highest probability and the RLBs of all other branches is checked. If this ratio is larger than a threshold *buff\_diff*, this branch will be included in set  $BS$ , because it offers a better RLB. If  $|BS| = 0$  no instruction is given.

**Step 2:** If  $|BS| \geq 1$ : Let  $P(b_k)$  denote the probability assigned to the  $k$ -th branch and  $MAX(P(b_k))$  denote the highest probability of all branches in  $FS$  (note:  $FS$  is a superset of  $BS$  and the probabilities of branches in both sets are the same). For each branch in  $BS$  check whether inequality 9 holds in which *prob\_diff* denotes a threshold for the ratio of the highest probability of all branches among  $FS$  and a given branch in  $BS$ .

$$\frac{MAX(P(b_k))}{P(b_k)} > prob\_diff \quad (9)$$

If this inequality holds, an instruction will be given, because chances are high that the agent misses a better RLB.

If none of these cases holds for the current junction, then no instruction is given and the agent chooses an option (step 7) based on the probabilities (step 5) assigned to each branch. With those conditions we try to predict costly mistakes rather than correct them because some mistakes can be expensive and unrecoverable regarding the goal of reaching the destination within a given distance threshold.

### Step 7: Choose an Option

At any junction agents either choose an option by following an instruction or by making a decision without having received an instruction. In the latter case, agents choose the branch they will continue on from the final selection set  $FS$  based on a weighted random choice. If an instruction was given, however, the same procedure as for the baseline applies (see above), although one important difference applies: The edge on which an agent traveled to the current junction is the only one which is excluded. Please note: Again, turn-around instructions are as well followed error-free.

This decision mechanism of giving an instruction remains the same for the whole route.

## 5 Results

In order to reduce the amount of combinations, a pre-test was done with 20 agents (out of 3000) and 15 routes (out of 100) using the following parameter values:

Maximum path length ( <i>max_dist</i> )	{1.2, 1.35, 1.5}
Buffer factor ( <i>buff_fact</i> )	{0.5, 0.6, 0.7, 0.8, 1.0}
Positive sum threshold ( <i>pos_sum</i> )	{0.5, 0.75}
Buffer difference threshold ( <i>buff_diff</i> )	{1.05, 1.2, 1.5}
Probability difference threshold ( <i>prob_diff</i> )	{1.05, 1.2, 1.5}
Already visited penalty factor ( <i>vis_pen</i> )	{0.1, 0.5, 0.75}

This yielded 810 different parameter combinations which were tried. Of those, for each city two parameter sets were chosen for the experiment, resulting in 6 different sets in total:

***best\_perc*** is the parameter set which yielded the maximum percentage of successful trials, thereby prioritizing the percentage of agents who arrive.

***best\_f\_ch*** is the parameter set which represents a trade off (found by multiplication) between the percentage of successful trials and the number of given instructions.

Each of the six parameter sets was used for each city (3000 agents and 100 routes) resulting in 18 runs, overall. Of these, the two best performing runs for each city were selected for the final analysis (see Table 1) applying again the *best\_perc* and *best\_f\_ch* criteria. There are, consequently, two datasets for every city for condition *free choice navigation*. Contrastingly, the baseline has only one dataset for each city as no additional parameters except *nav\_instr* need to be set. An overview of the results is presented in Table 2. Generally speaking, more people arrive with the baseline condition across all cities. The parameter set for *best\_f\_ch* leads to a similar number of instructions per traversed node as for the baseline.

■ **Table 1** The best parameter set for every city regarding *best\_perc* (%) and *best\_f\_ch* (% \* *f\_c*). D – Djibouti, M – Mexico, V – Vienna.

City	Best at	max dist	buff fact	pos sum	buff diff	prob diff	vis pen
D	%	1.5	0.5	0.75	1.2	1.2	0.5
M, V	%	1.5	0.5	0.75	1.5	1.5	0.1
D, M, V	% * <i>f_c</i>	1.5	0.7	0.75	1.5	1.05	0.5

Next, we will present the analysis regarding our hypotheses. In order to analyse differences between both conditions bootstrapping ( $B = 10000$  runs) was used and 95% percentile-based confidence intervals (CIs) are reported in square brackets. We refrain from calculating statistical tests due to the very large sample size. While there is a single dataset in the baseline condition, for our *free choice navigation* approach the relevant dataset (*best\_perc* or *best\_f\_ch*) is chosen as appropriate.

In the following, we detail several hypotheses with respect to our approach and provide the results of our analysis.

**Reduced Number of Navigation Instructions (H1)** As described above (see Section 1), *free choice navigation* approach equips wayfinders with flexibility in terms of route choice. We, therefore, hypothesize that people who reach the destination within a distance of  $1.5 \times sp\_len$  using our approach will receive less route instructions as compared to the baseline scenario.

■ **Table 2** Comparison of baseline (B) and *free choice navigation* (FCN) for agents who arrived within a certain percentage (leftmost column) of the shortest path length being allowed to walk 150% of the shortest path. Parameter sets for condition *free choice navigation* are given in Table 1 ( $max\_dist=1.5$ ). *base*: dataset of baseline; *best\_perc/best\_f\_ch*: datasets of *free choice navigation*; %: percentage of people who arrived; ipn: mean no. of instructions per traversed junction (ipn).

City	Djibouti						Mexico						Vienna					
Con.	B			FCN			B			FCN			B			FCN		
Dset	base		<i>best_perc</i>	<i>best_f_ch</i>		base		<i>best_perc</i>	<i>best_f_ch</i>		base		<i>best_perc</i>	<i>best_f_ch</i>				
Feat.	%	ipn	%	ipn	%	ipn	%	ipn	%	ipn	%	ipn	%	ipn	%	ipn	%	ipn
110%	0.74	0.37	0.13	0.66	0.04	0.27	0.81	0.32	0.13	0.62	0.06	0.30	0.77	0.39	0.12	0.61	0.03	0.20
120%	0.9	0.39	0.30	0.68	0.11	0.28	0.95	0.33	0.32	0.66	0.16	0.31	0.92	0.4	0.34	0.64	0.12	0.24
130%	0.96	0.39	0.51	0.7	0.24	0.31	0.98	0.33	0.55	0.69	0.30	0.33	0.97	0.40	0.59	0.67	0.25	0.28
140%	0.98	0.4	0.76	0.72	0.45	0.36	0.99	0.33	0.78	0.72	0.52	0.37	0.99	0.41	0.83	0.69	0.48	0.34
150%	0.99	0.4	0.91	0.74	0.81	0.43	1.0	0.33	0.9	0.74	0.81	0.43	1.0	0.41	0.95	0.71	0.85	0.41

Our figures do not support this hypothesis: For all cases, the mean difference in instructions given per traversed junction<sup>4</sup> between the baseline and our approach (*best\_f\_ch*) is negative, i.e., less instructions were given in the baseline condition: For Djibouti ( $M = -0.037$ ,  $SD = 0.0004$ ,  $[-0.0379; -0.0361]$ ) our approach yielded on average one additional instruction every 27 ( $\frac{1}{mean}$ ) junctions, whereas in Mexico ( $M = -0.103$ ,  $SD = 0.0004$ ,  $[-0.104; -0.102]$ ) this value increases to every 10 junctions. The smallest difference between the two conditions was found for Vienna ( $M = -0.004$ ,  $SD = 0.0004$ ,  $[-0.005; -0.003]$ ) where one additional instruction every 250 junctions is expected.

**Longer Routes Enable More Free Choices (H2)** Based on the fact that longer routes result in an increased maximum absolute route length and in line with H1, we also assume that successful agents will have made a higher number of free choices on longer routes. Similar to H1, our results (*best\_f\_ch*) do not support this hypothesis. We found a weak negative Spearman correlation between the number of junctions traversed and the % of free choices of all decisions an agent made across cities. Djibouti ( $M = -0.168$ ,  $SD = 0.002$ ,  $[-0.172; -0.164]$ ) and Vienna ( $M = -0.179$ ,  $SD = 0.002$ ,  $[-0.183; -0.175]$ ) showed a stronger correlation than Mexico ( $M = -0.119$ ,  $SD = 0.002$ ,  $[-0.123; -0.116]$ ).

**Percentage of People Reaching Destination (H3)** The *free choice navigation* we suggest imposes increased cognitive load on wayfinders: They have to make spatial decisions based on path integration and their belief about the destination vector (see Section 2). Therefore, we hypothesize that, within a given maximum distance threshold (150%), more people will arrive at the destination with the baseline than with our *free choice navigation* approach.

Our data is in line with this assumption. Again, the difference between the baseline and *free choice navigation* (*best\_perc*) is reported: In Djibouti ( $M = 0.085$ ,  $SD = 0.0005$ ,  $[0.084; 0.086]$ ) on average 8.5 percentage points less trials ended successfully. In Mexico ( $M = 0.100$ ,  $SD = 0.0005$ ,  $[0.099; 0.102]$ ) the figures show 10 percentage points. For Vienna ( $M = 0.047$ ,  $SD = 0.0004$ ,  $[0.046; 0.048]$ ), again, the smallest difference compared to the baseline was found, showing an average difference of 4.7 percentage points.

**Low Spatial Abilities of Failures (H4)** Based on the need to make their own spatial decisions (see also H2), we assume that the fraction of people, who is not able to arrive at the destination within a threshold of  $1.5 \times sp\_length$ , will be highest within those agents who show low spatial abilities.

<sup>4</sup> The number of obtained instructions was normalized by the number of traversed nodes as *free choice navigation* agents walked on average more.

■ **Table 3** Numbers regarding failed trials: a) Share of failed trials per city and ability category. b) Share of agents who failed at least once.

(a) The share of failed trials within a 150% distance threshold categorized by city and env. spatial abilities.

City	Low	Medium	High
Djibouti	25.95	6.95	1.34
Mexico	28.99	7.68	0.79
Vienna	16.47	3.46	0.23

(b) Share, mean (M) and standard deviation (SD) of agents for a given category of env. spatial abilities **who failed one or more trials**. *low* (521): [0; 0.3]; *medium* (2028): (0.3; 0.7]; *high* (451): [0.7; 1]

City	Low			Medium			High		
	Feat.	%	M   SD	%	M   SD	%	M   SD		
D	100%	.2	.079	99%	.50	.11	79%	.79	.07
M	100%	.2	.079	98%	.50	.11	51%	.77	.06
V	100%	.2	.079	86%	.49	.10	20%	.75	.05

In order to investigate this hypothesis, agents are categorized, based on their *env\_sp\_ab* and its standard deviation, into 3 groups: low ([0;0.3],  $N_{low} = 521$ ), medium ((0.3;0.7],  $N_{med} = 2028$ ) and high ((0.7;1],  $N_{high} = 451$ ). The data obtained by the simulation (*best\_perc*) supports our assumption: Across cities (*best\_perc*) agents with *low env\_sp\_ab* show the highest share of failed trials (see Table 3a). Moreover, each agent with low spatial abilities failed at least on one trial (see Table 3b). In Vienna 20% of high-level agents did not reach their destination *at least once*, whereas in Djibouti and Mexico this share was 79% and 51%, respectively.

**Higher Spatial Abilities Yield Shorter Routes (H5)** Based on prior evidence on path integration (see e.g., [24]) we assume that wayfinders with high spatial abilities will be able to take shorter routes, in terms of deviation from shortest path length.

There is a strong negative correlation between *env\_sp\_ab* and the path length (*best\_perc*); which supports this hypothesis. Both, Djibouti ( $M = -0.50$ ,  $SD = 0.002$ , [-0.504; -0.498]) and Mexico ( $M = -0.507$ ,  $SD = 0.002$ , [-0.51; -0.504]) show similar correlations, whereas Vienna ( $M = -0.566$ ,  $SD = 0.001$ , [-0.568; -0.563]) shows a stronger correlation.

## 6 Discussion and Limitations

We will discuss the results of our agent-based simulation along two lines. First, we will discuss our findings and contributions with respect to the idea of *free choice navigation*. Second, we will continue with respect to the plausibility of the model and its limitations.

### 6.1 Discussion

Using TBT navigation systems, users are required to follow a predefined route including predefined turns. This system behavior, however, results in a reduced interaction between users and the spatial environment, ultimately leading to (potentially) adverse effects in terms of spatial orientation. The primary goal of this paper is, therefore, to propose the concept of *free choice navigation* and initially test this assistance approach by means of a simulation study. Our approach has the potential to remedy these effects and is expected to foster spatial knowledge acquisition as it aims to give user more freedom and, at the same time, ensures reasonable route lengths.

Generally speaking, plausibility and validity are both important concepts in simulation studies [5]. While the former can be judged according to specific figures, the latter requires empirical evidence. Checking validity must, hence, be left for future work (see Section 7); our figures, however, indicate the plausibility of agent behavior. Our *free choice navigation* approach is plausible as cities with different morphologies [32] yield different results (see Tab. 2) which is also in line with prior evidence (see e.g., [4]). This difference is also reflected in our results regarding H1: Our approach does not yield less navigation instructions per

traversed node compared to the baseline system. While the effect can be neglected for Vienna (average route length: 87.8 junctions, i.e., 0-1 instructions more than the baseline system) and is of low relevance for Djibouti (mean length: 67.5 junctions, i.e., 2-3 instructions more), a considerable increase (mean length: 70.6 junctions, i.e., 7 instructions more) can be seen for Mexico City, where 10% more instructions are needed than in the baseline condition. The reasons as well as the impact on user experience leaves much room for further research.

The plausibility of our model is further supported by our results regarding H3, H4, H5: Less people arrived with our approach than in the baseline condition (H3), agents with low environmental spatial abilities fail more often (H4) and the higher these abilities the shorter the path taken (H5). In addition to that, each agent which has low abilities does not reach its destination at least once (see Table 3b). All of these results are in line with our expectations and, taken together, suggest that environmental spatial abilities play a key role (see future work below). Having said this, the simulation results for our *free choice navigation* approach yield, moreover, promising results with respect to success rate (i.e., reaching the destination within  $1.5 \times \text{len}(\text{shortest\_path})$ ) and distance traveled: In each city more than 90% of all agents arrive within this threshold. More importantly, on average this upper limit was not reached at all (Djibouti:  $M = 1.26$ ,  $SD = 0.13$ ; Mexico:  $M = 1.25$ ,  $SD = 0.127$ ; Vienna:  $M = 1.25$ ,  $SD = 0.121$ ). A mean detour of about 25% seems reasonable in a leisure scenario and our approach yields shorter average distances than reported in other, vibro-tactile based, beeline studies (see e.g., [29] in which mean distances greater than  $1.5 \times \text{len}(\text{shortest\_path})$  are reported). While the number of route instructions issued is not less than for the baseline system the *free choice navigation* approach proposed allows for a particularly high share of free choices (Djibouti: 0.57, Mexico: 0.57, Vienna: 0.59) which results in a lot of engagement with the spatial environment traversed. Having said this, however, our results do not support hypothesis H2: We found a weak negative correlation between the number of traversed intersections and the share of free choices along a route (i.e., the number of intersections at which no route instruction is given): One possible reason for that can be that the parameter set produces an artifact in the data, hence, further research is needed. Furthermore, the higher overall success rate and the lower share of agents with medium and high environmental spatial abilities which did not arrive at least once (see Table 3b) suggest Vienna to be easier to navigate than Djibouti and Mexico City.

Taken together, this discussion reveals that the *free choice navigation* is reasonable. At the same time, the chosen parameters are crucial for the figures achieved. Clearly, our model is not optimal and can, therefore, act as a baseline other researchers can compete with. There are several possibilities for improvement: For example, reducing the number of instructions or increasing the percentage of arrived agents. As a consequence, the data will be published at <https://geoinfo.geo.tuwien.ac.at/resources/>

## 6.2 Limitations

Despite the promising results, several limitations apply. On the one hand, the environment can be modeled more complex considering for example junction geometry [11] or environmental data like building footprints or points of interests. On the other hand, a more elaborate modeling of agents is feasible, in particular with respect to interpretation of route instructions (of different types) [34] or by considering further wayfinding preferences [3]. A further limitation of our work is related to the application of the introduced cone (see Section 2) for real scenarios. We assumed to know the approximate direction that agents would follow (i.e., the direction of the cone). To apply this to humans, it is necessary to know the direction they will follow. Recent research in human activity recognition has shown very positive results (e.g., [38, 2]), which can be used for our purpose to determine the direction of the cone.

## 7 Conclusion and Future Work

We introduced *free choice navigation*, a novel approach with the intention to increase the freedom of choice during navigation. This approach was evaluated by means of an agent-based simulation study and compared against a TBT approach, serving as a baseline. The agent-based model used for the simulations is a further contribution, introducing the concept of the *cone*, which encodes the user's spatial abilities and confidence. Our findings are in line with our expectations concerning the proportion of free choices during navigation as well as the impact of spatial abilities on the effectiveness of our approach. Furthermore, the results confirm the plausibility of the introduced agent-based model.

Future human subject experiments in real-world environments are required in order to address a series of open research questions. First, the validity of the presented agent-based model could be investigated by comparing the simulated results with the ones obtained from humans in a real environment. Furthermore, we expect that the *free choice navigation* will foster spatial knowledge acquisition due to the increased engagement with the environment. Along the same line, also aspects concerning user experience, cognitive load, or uncertainty should be addressed in human subject experiments. The data and model will be made available and can serve as a baseline for further development. The model and the results can, thus, be used by the community as a benchmark for future iterations of the model.

---

### References

- 1 S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. O'Hare. Agent based modelling and simulation tools: A review of the state-of-art software. *Comp. Science Review*, 24:13–33, 2017.
- 2 N. Alinaghi, M. Kattenbeck, A. Golab, and I. Giannopoulos. Will You Take this Turn? Gaze-Based Turning Activity Recognition During Navigation. In K. Janowicz and J. A. Verstegen, editors, *11th Int. Conf. on Geographic Information Science (GIScience 2021) - Part II*, 2021.
- 3 J. N. Bailenson, M. S. Shum, and D. H. Uttal. Road climbing: Principles governing asymmetric route choices on maps. *J. of Env. Psychology*, 18(3):251–264, 1998.
- 4 D. Bhowmick, S. Winter, and M. Stevenxon. Comparing the costs of pedestrian wayfinding heuristics across different urban network morphologies. *GeoComputation*, pages 1–15, 2019.
- 5 L. Crociani, G. Vizzari, D. Yanagisawa, K. Nishinari, and S. Bandini. Route choice in pedestrian simulation: Design and evaluation of a model based on empirical observations. *Intelligenza Artificiale*, 10(2):163–182, 2016.
- 6 L. Dahmani and V. D. Bohbot. Habitual use of GPS negatively impacts spatial memory during self-guided navigation. *Scientific Reports*, 10(1):1–14, 2020.
- 7 N. Dethlefs, Y. Wu, A. Kazerani, and S. Winter. Generation of adaptive route descriptions in urban environments. *SPAT COGN COMPUT*, 11(2):153–177, 2011.
- 8 D. Dobbstein, E. Rukzio, and P. Henzler. Unconstrained pedestrian navigation based on vibro-tactile feedback around the wristband of a smartwatch. *Proc. of CHI '16*, p.2439–2445, 2016.
- 9 R.M. Downs and D. Stea. *Maps in Minds: Reflections on Cognitive Mapping*. Geography Series. Harper & Row, 1977.
- 10 G. Filomena, E. Manley, and J. A. Verstegen. Perception of urban subdivisions in pedestrian movement simulation. *PLoS ONE*, 15(12):1–27, 2020.
- 11 P. Fogliaroni, D. Bucher, N. Jankovic, and I. Giannopoulos. Intersections of Our World. In Winter et al., editor, *10th Int. Conf. on Geographic Information Science*, 2018.
- 12 D. Gallo, S. Shreepriya, and J. Willamowski. RunAhead: Exploring Head Scanning based Navigation for Runners. In *Proc. of CHI 2020.*, pages 1–13, 2020.
- 13 A. Gardony, T. Brunyé, and H. Taylor. Navigational Aids and Spatial Memory Impairment: The Role of Divided Attention. *SPAT COGN COMPUT*, 15(4):246–284, 2015.

- 14 I. Giannopoulos, P. Kiefer, and M. Raubal. Gazenav: Gaze-based pedestrian navigation. In *Proc. of MobileHCI 2015*, page 337–346. ACM, 2015.
- 15 I. Giannopoulos, P. Kiefer, M. Raubal, K. F. Richter, and T. Thrash. Wayfinding decision situations: A conceptual model and evaluation. In *Int. Conf. on Geographic Information Science*, pages 221–234. Springer, 2014.
- 16 K. Gramann, P. Hoepner, and K. Karrer-Gauss. Modified navigation instructions for spatial navigation assistance systems lead to incidental spatial learning. *FRONT PSYCHOL*, 8, 2017.
- 17 A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proc. of the 7th Python in Science Conf.*, pages 11–15, Pasadena, CA USA, 2008.
- 18 S. K. Harootonian, R. C. Wilson, L. Hejtmánek, E. M. Ziskin, and A. D. Ekstrom. Path integration in large-scale space and with novel geometries: Comparing vector addition and encoding-error models. *PLoS computational biology*, 16(5):e1007489, 2020.
- 19 H. Hochmair. Investigating the effectiveness of the least-angle strategy for wayfinding in unknown street networks. *Env and Planning B: Planning and Design*, 32(5):673–691, 2005.
- 20 H. Hochmair and A. Frank. Influence of estimation errors on wayfinding-decisions in unknown street networks – analyzing the least-angle strategy. *SCC*, 2(4):283–313, 2002.
- 21 T. Ishikawa. Satellite Navigation and Geospatial Awareness: Long-Term Effects of Using Navigation Tools on Wayfinding and Spatial Orientation. *Prof. Geographer*, 71(2), 2019.
- 22 D. Jonietz and P. Kiefer. Uncertainty in Wayfinding: A Conceptual Framework and Agent-Based Model. In E. Clementini et al., editor, *Proc. of COSIT 2017*, pages 15:1–15:14, 2017.
- 23 Angelika Kneidl. *Methoden zur Abbildung menschlichen Navigationsverhaltens bei der Modellierung von Fußgängerströmen*. Dissertation, Technische Universität München, 2013.
- 24 J. M. Loomis, R. L. Klatzky, R. G. Golledge, and J. W. Philbeck. Human navigation by path integration. In *Wayfinding Behavior*, pages 125–151. Johns Hopkins University Press, 1999.
- 25 C. Magnusson, K. Rasmus-Gröhn, and D. Szymczak. Navigation by pointing to GPS locations. *Personal and Ubiquitous Computing*, 16(8):959–971, 2012.
- 26 M. Pielot and S. Boll. Tactile wayfinder: Comparison of tactile waypoint navigation with commercial pedestrian navigation systems. In P. Floréen, A. Krüger, and M. Spasojevic, editors, *Proc. of the 8th Int. Conf. on Pervasive Computing*, pages 76–93. Springer Berlin Heidelberg, 2010.
- 27 A. Rao and M. Georgeff. BDI Agents: From Theory to Practice. *Proc. of ICMAS '95*, pages 312–319, 1995.
- 28 K. F. Richter and M. Duckham. Simplest instructions: Finding easy-to-describe routes for navigation. In T. J. Cova, H. J. Miller, K. Beard, A. U. Frank, and M. F. Goodchild, editors, *Geographic Information Science*, pages 274–289, Berlin, Heidelberg, 2008. Springer.
- 29 S. Robinson, M. Jones, P. Eslambolchilar, R. Murray-Smith, and M. Lindborg. “I Did It My Way”: Moving Away from the Tyranny of Turn-by-Turn Pedestrian Navigation. In *Proc. of MobileHCI '10*, pages 341–344, 2010.
- 30 G.-L. Savino, L. Meyer, E. Schade, T. Tenbrink, and J. Schöning. Point me in the right direction: Understanding user behaviour with as-the-crow-flies navigation. In *Proc. of MobileHCI '20*, pages 1–11, 2020.
- 31 F. Shatu, T. Yigitcanlar, and J. Bunker. Shortest path distance vs. least directional change: Empirical testing of space syntax and geographic theories concerning pedestrian route choice behaviour. *J. of Transport Geography*, 74(May 2018):37–52, 2019.
- 32 J. Thompson, M. Stevenson, and J. S. Wijnands et al. (8). A global analysis of urban design types and road transport injury: an image processing study. *The Lancet Planetary Health*, 4(1):e32–e42, 2020.
- 33 P. W. Thorndyke and B. Hayes-Roth. Differences in spatial knowledge acquired from maps and navigation. *Cognitive psychology*, 14(4):560–589, 1982.
- 34 L. B. Tschander, H. R. Schmidtke, C. Eschenbach, C. Habel, and L. Kulik. A geometric agent following route instructions. In *Int. Conf. on Spatial Cognition*, pages 89–111. Springer, 2002.

## 9:16 Navigating Your Way! Increasing the Freedom of Choice During Wayfinding

- 35 J. M. Usher and L. Strawderman. Simulating operational behaviors of pedestrian navigation. *Computers & Industrial Engineering*, 59(4):736–747, 2010.
- 36 A. Wunderlich and K. Gramann. Electrocortical evidence for long-term incidental spatial learning through modified navigation instructions. In *Proc. of Spatial Cognition '18*, 2018.
- 37 D. Xu, X. Huang, J. Mango, X. Li, and Z. Li. Simulating multi-exit evacuation using deep reinforcement learning. *Transactions in GIS*, pages 1–23, 2021.
- 38 J. Ye, X. Li, X. Zhang, Q. Zhang, and W. Chen. Deep learning-based human activity real-time recognition for pedestrian navigation. *Sensors*, 20(9), 2020.
- 39 H. D. Zimmer, S. Münzer, and J. Baus. *From Resource-Adaptive Navigation Assistance to Augmented Cognition*, pages 35–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

# Terrain Prickliness: Theoretical Grounds for High Complexity Viewsheds

**Ankush Acharyya** ✉

The Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic

**Ramesh K. Jallu** ✉

The Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic

**Maarten Löffler** ✉

Department of Information and Computing Sciences, Utrecht University, The Netherlands

**Gert G.T. Meijer** ✉

Academy of ICT and Creative Technologies,  
NHL Stenden University of Applied Sciences, The Netherlands

**Maria Saumell** ✉

The Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic  
Department of Theoretical Computer Science, Faculty of Information Technology,  
Czech Technical University in Prague, Czech Republic

**Rodrigo I. Silveira** ✉

Department of Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain

**Frank Staals** ✉

Department of Information and Computing Sciences, Utrecht University, The Netherlands

---

## Abstract

An important task in terrain analysis is computing *viewsheds*. A viewshed is the union of all the parts of the terrain that are visible from a given viewpoint or set of viewpoints. The complexity of a viewshed can vary significantly depending on the terrain topography and the viewpoint position.

In this work we study a new topographic attribute, the *prickliness*, that measures the number of local maxima in a terrain from all possible angles of view. We show that the prickliness effectively captures the potential of terrains to have high complexity viewsheds. We present near-optimal algorithms to compute it for TIN terrains, and efficient approximate algorithms for raster DEMs. We validate the usefulness of the prickliness attribute with experiments in a large set of real terrains.

**2012 ACM Subject Classification** Theory of computation → Theory and algorithms for application domains

**Keywords and phrases** Digital elevation model, Triangulated irregular network, Viewshed complexity

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.10

**Related Version** *Full Version*: <https://arxiv.org/abs/2103.06696>

## Supplementary Material

*Software (Source Code)*: <https://github.com/GTMeijer/Prickliness>  
archived at `swh:1:dir:c360f8c5b838bfe88910d26aad151dee69f69364`

*Software (Source Code)*: [https://github.com/GTMeijer/TIN\\_Viewsheds](https://github.com/GTMeijer/TIN_Viewsheds)  
archived at `swh:1:dir:911b84528046c62ddd56c32905926748dd59791e`

**Funding** This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734922. A. A., R. J., and M. S. were supported by the Czech Science Foundation, grant number GJ19-06792Y, and by institutional support RVO: 67985807. M. L. was partially supported by the Netherlands Organization for Scientific Research (NWO) under project no. 614.001.504. R. S. was supported by AEI, project PID2019-104129GB-I00/AEI/10.13039/501100011033, and Gen. Cat. DGR 2017SGR1640.



© Ankush Acharyya, Ramesh K. Jallu, Maarten Löffler, Gert G.T. Meijer, Maria Saumell, Rodrigo I. Silveira, and Frank Staals;  
licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 10; pp. 10:1–10:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

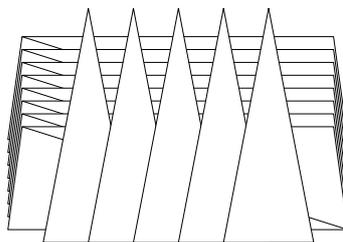
**Acknowledgements** The authors would like to thank Jeff Phillips for a stimulating discussion that, years later, led to the notion of prickliness.

## 1 Introduction

Digital terrain models represent part of the earth’s surface, and are used to solve a variety of problems in geographic science. An important task is viewshed analysis: determining which parts of a terrain are visible from certain terrain locations. Two points  $p$  and  $q$  are mutually *visible* if the line of sight defined by line segment  $\overline{pq}$  does not intersect the interior of the terrain. Given a *viewpoint*  $p$ , the *viewshed* of  $p$  is the set of all terrain points that are visible from  $p$ . Similarly, the viewshed of a set of viewpoints  $P$  is defined as the set of all terrain points that are visible from *at least* one viewpoint in  $P$ . Viewsheds are useful, for example, in evaluating the visual impact of potential constructions [2], analyzing the coverage of an area by fire watchtowers [12], or measuring the scenic beauty of a landscape [1, 21].

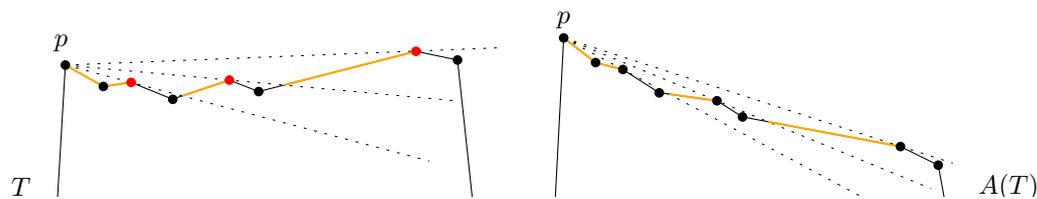
**Discrete and continuous terrain representations.** Two major terrain representations are prevalent in GIS. The simplest and most widespread is the raster, or *digital elevation model* (DEM), consisting of a rectangular grid where each cell stores an elevation.<sup>①</sup> The main alternative is a vector representation, or *triangulated irregular network* (TIN), where a set of irregularly spaced elevation points are connected into a triangulation. A TIN can be viewed as a continuous  $xy$ -monotone polyhedral surface in  $\mathbb{R}^3$ . A viewshed in a DEM is the set of all raster cells that are visible from at least one viewpoint. In contrast, a viewshed in a TIN is the union of all parts of triangles that are visible from at least one viewpoint.

DEMs are simpler to analyze than TINs and facilitate most analysis tasks. The main advantage of TINs is that they require less storage space. Both models have been considered extensively in the literature for viewshed analysis, see Dean [4] for a complete comparison of both models in the context of forest viewshed. Some studies suggest that TINs can be superior to DEMs in viewshed computations [4], but experimental evidence is inconclusive [19]. This is in part due to the fact that the viewshed algorithms used in [19] do not compute the visible part of each triangle, but only attempt to determine whether each triangle is completely visible. This introduces an additional source of error and does not make use of all the information contained in the TIN.



■ **Figure 1** Part of a TIN with a high-complexity viewshed. The viewpoint (not shown) is placed at the center of projection. The relevant triangles of the TIN are the ones shown, which define  $n$  peaks and ridges. The viewshed in this case is formed by  $\Theta(n^2)$  visible regions.

<sup>①</sup> For the sake of simplicity, in this paper we use DEM to denote the raster version of a DEM.



■ **Figure 2** Left: a TIN (1.5D profile) with three peaks and one viewpoint ( $p$ ), with a viewshed composed of three parts (visible parts shown orange). Right: transformation of the terrain with no peaks (other than  $p$ ) but the same viewshed complexity. Dotted segments show lines of sight from  $p$ .

**Viewshed complexity.** The algorithmic study of viewsheds focuses on two main aspects: the complexity of the viewsheds, and their efficient computation. In this work, we are interested in their complexity. We use the information-theoretic meaning of “complexity”: the complexity of an object is the number of bits needed to represent it in memory. Therefore, in the case of TINs, viewshed complexity is defined as the total number of vertices of the polygons that form the viewshed. In the case of DEMs, there are several ways to measure viewshed complexity. To facilitate comparison between TIN and DEM viewsheds, we convert the visible areas in the raster viewshed to polygons, and define the viewshed complexity as the total number of vertices in those polygons. A typical high-complexity viewshed construction for a TIN is shown schematically in Fig. 1, where one viewpoint would be placed at the center of projection, and both the number of vertical and horizontal triangles is  $\Theta(n)$ , for  $n$  terrain vertices. The vertical peaks form a grid-like pattern with the horizontal triangles, leading to a viewshed with  $\Theta(n^2)$  visible triangle pieces.

While a viewshed can have high complexity, this is expected to be uncommon in real terrains [3]. There have been attempts to define theoretical conditions for a (TIN) terrain that guarantee, among others, that viewsheds cannot be that large. For instance, Moet et al. [18] showed that if terrain triangles satisfy certain “realistic” shape conditions, viewsheds have  $O(n\sqrt{n})$  complexity. De Berg et al. [3] showed that similar conditions guarantee worst-case expected complexity of  $\Theta(n)$  when the vertex heights are subject to uniform noise.

**Viewsheds and peaks.** The topography of the terrain has a strong influence on the potential complexity of the viewshed. To give an extreme example, in a totally concave terrain, the viewshed of any viewpoint will be the whole terrain, and has a trivial description. Intuitively, to obtain a high complexity viewshed as in Fig. 1, one needs a large number of obstacles obstructing the visibility from the viewpoint, which requires a somewhat rough topography.

In fact, it is well-established that viewsheds tend to be more complex in terrains that are more “rugged” [13]. This leads to the natural question of which terrain characteristics correlate with high complexity viewsheds. Several topographic attributes have been proposed to capture different aspects of the roughness of a terrain, such as the *terrain ruggedness index* [20], the *terrain shape index* [16], or the *fractal dimension* [14]. These attributes focus on aspects like the amount of elevation change between adjacent parts of a terrain, its overall shape, or the terrain complexity. However, none of them is specifically intended to capture the possibility to produce high complexity viewsheds, and there is no theoretical evidence for such a correlation. Moreover, these attributes are locally defined, and measure only attributes of the local neighborhood of one single point. While we can average these measures over the whole terrain, given the global nature of visibility, it is unclear a priori whether such measures are suitable for predicting viewshed complexity. We refer to Dong et al. [5] for a systematic classification of topographic attributes.

One very simple and natural global measure of the ruggedness of a terrain that is relevant for viewshed complexity is to simply count the number of *peaks* (i.e., local maxima) in the terrain. It has been observed in the literature that areas with higher elevation difference, and hence, more peaks, cause irregularities in viewsheds [9, 12], and this idea aligns with our theoretical understanding: the quadratic example from Fig. 1 is designed by creating an artificial row of peaks, and placing a viewpoint behind them. However, while it seems reasonable to use the peak count as complexity measure, there is no theoretical correlation between the number of peaks and the viewshed complexity. This is easily seen by performing a simple trick: any terrain can be made arbitrarily flat by scaling it in the  $z$ -dimension by a very small factor, and then rotating it slightly – this results in a valid terrain without any peaks, but retains the same viewshed complexity. See Fig. 2 for an example in 1.5D. In fact, viewshed complexity is invariant under affine transformations (i.e., scalings, rotations, and translations) of the terrain: the application of any affine combination to the terrain and the viewpoints results in a viewshed of the same complexity. Hence, any measure that has provable correlation with it must be affine-invariant as well. This is a common problem to establish theoretical guarantees on viewshed complexity, or to design features of “realistic” terrains in general [3, 18]. In fact, it is easy to see that none of the terrain attributes mentioned above is affine-invariant.

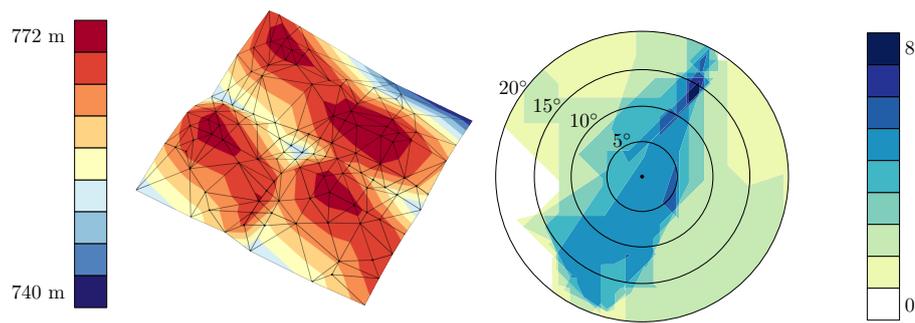
**Prickliness.** In this work we propose a new topographic attribute: the *prickliness*. The definition follows directly from the above observations: it counts the number of peaks in a terrain, but does so *for every possible affine transformation* of the terrain. We first present a definition for TINs, and then we explain how the definition carries over to DEMs.

Let  $T$  be a triangulated surface that is  $xy$ -monotone. Let  $A$  be an affine transformation, and let  $A(T)$  be the terrain obtained after applying  $A$  to  $T$ . We define  $m(A(T))$  to be the number of internal and convex vertices of  $T$ <sup>②</sup> that are peaks in  $A(T)$ . Let  $\mathcal{A}(T)$  be the set of all affine transformations of  $T$ . We define the *prickliness* of  $T$ ,  $\pi(T)$ , to be the maximum number of local maxima over all transformations of  $T$ ; that is,  $\pi(T) = \max_{A \in \mathcal{A}(T)} m(A(T))$ . We start by observing that, essentially, the prickliness considers all possible *directions* in which the number of peaks are counted. Let  $\vec{v}$  be a vector in  $\mathbb{R}^3$ . Let  $\pi_{\vec{v}}(T)$  be the number of internal and convex vertices of  $T$  that are local maxima of  $T$  in direction  $\vec{v}$ ; that is, the number of internal and convex vertices of  $T$  for which the local neighborhood does not extend further than that vertex in direction  $\vec{v}$ .

► **Observation 1.**  $\pi(T) = \max_{\vec{v}} \pi_{\vec{v}}(T)$

Using this observation, we reduce the space of all affine transformations to the 2-dimensional space of all directions in 3D. Since  $T$  is a terrain, for any  $\vec{v}$  with a negative  $z$ -coordinate we have  $\pi_{\vec{v}}(T) = 0$  by definition, thus the interesting directions reduce to the points on the (positive) unit half-sphere. This provides a natural way to visualize the prickliness of a terrain. Each direction can be expressed using two angles  $\theta$  and  $\phi$  (i.e., using spherical coordinates), where  $\theta$  represents the polar angle and  $\phi$  the azimuthal angle. Fig. 3 shows a small terrain and the resulting prickliness, showing a projection of the half-sphere, where each point represents a direction, and its color indicates its prickliness. Note that we specifically define prickliness to be the *maximum* over all orientations rather, say, the average over all orientations. Even for a terrain with high-complexity viewsheds like the one

<sup>②</sup> We explicitly only count vertices that are already convex in the *original* terrain, since some affine transformations will transform local minima / concave vertices of the original terrain into local maxima.



■ **Figure 3** (left) A TIN  $T$ , with triangulation edges shown in black, and elevation indicated using colors. (right) A visualization of the prickliness of  $T$  as a function of the angles  $(\theta, \phi)$  that define each direction (circles indicate contour lines for  $\theta$ ); color indicates prickliness. The maximum prickliness is 8, attained at a direction of roughly  $\theta = 13^\circ$  and  $\phi = 60^\circ$  (north-east from the origin).

in Fig. 1, the average number of peaks would still be relatively small since there are many orientations with a small number of peaks. Hence, such a definition is unlikely to accurately capture the complexity of viewsheds on a terrain.

We notice that all previous notions easily translate to DEMs. The centers of the DEM cells can be seen as the *vertices* of the terrain, and every internal vertex of the terrain has eight neighbors given by the cell centers of the eight neighboring cells. Hence, in the definitions for DEMs, the notion of *adjacent vertices* for TINs is replaced by that of *neighbors*. A vertex is a local maximum for some affine transformation of the terrain if all of its neighbors have a lower or equal  $z$ -coordinate. This gives an equivalent definition of  $\pi(T)$  when  $T$  is a DEM.

**Results and organization.** The remainder of this paper is organized as follows.

- In Section 2, we review existing topographic attributes in more detail, and discuss how attributes defined for DEMs can be adapted to TINs.
- We show in Section 3 that the viewshed complexity of a single viewpoint in a TIN terrain cannot be higher than  $O(n \cdot \pi(T))$ , and this is tight.
- In Section 4, we consider the question of how to compute prickliness. We provide an  $O(n^2)$  time algorithm for TIN terrains, which is near-optimal, and an efficient approximate algorithm for DEM terrains.
- In Sections 5-7, we report on experiments that measure the values of distinct topographic attributes (including the prickliness) of real terrains, and analyze their possible correlation with viewshed complexity. From the experiments, we conclude that prickliness provides such a correlation in the case of TIN terrains, while the other measures perform more poorly. The situation for DEM terrains is less clear.
- Finally, we provide our code implementing two key algorithms for this work: an algorithm to calculate the prickliness of a TIN terrain and an algorithm to calculate the combined viewshed originating from a set of multiple viewpoints.

## 2 Topographic attributes

In addition to prickliness, we consider the following topographic attributes, and their relation to viewshed complexity.

**Terrain Ruggedness Index (TRI).** The Terrain Ruggedness Index measures the variability in the elevation of the terrain [20]. Riley et al. originally defined TRI specifically for DEM terrains as follows. Let  $c$  be a cell of the terrain, and let  $\mathcal{N}(c)$  denote the set of (at most) eight neighboring cells of  $c$ . The TRI of  $c$  is then defined as

$$pTRI(c) = \sqrt{\frac{1}{|\mathcal{N}(c)|} \sum_{q \in \mathcal{N}(c)} (c_z - q_z)^2},$$

where  $|\mathcal{N}(c)|$  denotes the cardinality of  $\mathcal{N}(c)$ . Hence,  $pTRI(c)$  essentially measures the standard deviation of the difference in height between  $c$  and the points in  $\mathcal{N}(c)$ . The Terrain Roughness Index  $TRI(T)$  of  $T$  is the average  $pTRI(T, c)$  value over all cells  $c$  in  $T$ .

For TINs, we have adapted the definition as follows:  $\mathcal{N}(c)$  is defined as the set of vertices which are adjacent to a given vertex  $c$  in  $T$ . The Terrain Roughness Index  $TRI(T)$  is then obtained as the average  $pTRI(T, c)$  value over all vertices  $c$  of  $T$ .

**Terrain Shape Index (TSI).** The Terrain Shape Index also measures the “shape” of the terrain [16]. Let  $\mathcal{C}(c, r, T)$  denote the intersection of  $T$  with a vertical cylinder of radius  $r$  centered at  $c$  (so after projecting all points to the plane, the points in  $\mathcal{C}(c, r, T)$  lie on a circle of radius  $r$  centered at  $c$ ). For ease of computation, we discretize  $\mathcal{C}(c, r, T)$ : for DEMs we define  $C(c, r, T)$  to be the grid cells intersected by  $\mathcal{C}(c, r, T)$ , and for TINs we define  $C(c, r, T)$  as a set of 360 equally spaced points on  $\mathcal{C}(c, r, T)$ . The TSI of a point  $c$  is then defined as

$$pTSI(c, r, T) = \frac{1}{r|C(c, r, T)|} \sum_{q \in C(c, r, T)} c_z - q_z,$$

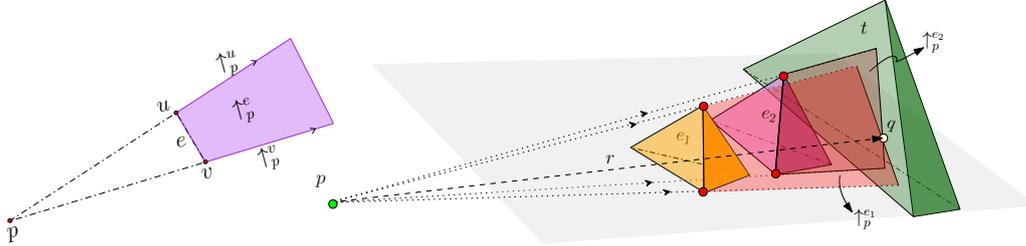
and essentially measures the average difference in height between “center” point  $c$  and the points at (planar) distance  $r$  to  $c$ , normalized by  $r$ . The Terrain Shape Index  $TSI(T, r)$  of the entire terrain  $T$  is the average  $pTSI(c, r, T)$  over all cells (in case of a DEM) or vertices (in case of a TIN) of  $T$ . We choose  $r = 1000\text{m}$  (which is roughly eight percent of the width of our terrains) in our experiments.

**Fractal dimension (FD).** The (local) fractal dimension measures the roughness around a point  $c$  on the terrain over various scales [14, 22]. We use the definition of Taud and Parrot [22] that uses a box-counting method, and is defined as follows. Let  $w$  be the width of a cell in the DEM, and let  $s \in \mathbb{N}$  be a size parameter. For  $q \in 1..s/2$ , consider subdividing the cube with side length  $sw$  centered at  $c$  into  $(s/q)^3$  cubes of side length  $qw$ . Let  $\mathcal{C}_s(c, q)$  denote the resulting set of cubes, and define  $N_s(c, q, T)$  as the number of cubes from  $\mathcal{C}_s(c, q)$  that contain a “unit” cube from  $\mathcal{C}_s(c, 1)$  lying fully below the terrain  $T$ . Let  $\ell_s(c, T)$  be the linear function that best fits (i.e. minimizes the sum of squared errors) the set of points  $\{(\ln(q), \ln(N_s(c, q, T))) \mid q \in 1..s/2\}$  resulting from those measurements. The fractal dimension  $pFD_s(c, T)$  at  $c$  is then defined as the inverse of the slope of  $\ell_s(c, T)$ . The fractal dimension  $FD(T, s)$  of the DEM terrain itself is again the average over all DEM cells. Following Taud and Parrot we use  $s = 24$  in our experiments. For our TIN terrains, we keep  $w$  the same as in their original DEM representations, and average over all vertices.

### 3 Prickliness and viewshed complexities

In this section we show that there is a provable relation between prickliness and viewshed complexity. We present our analysis for TINs, since they provide a natural way to measure viewshed complexity. However, a similar analysis can be carried out for DEMs as well.

Let  $v$  be a vertex of  $T$ , and let  $p$  be a viewpoint. We denote by  $\uparrow_p^v$  the half-line with origin at  $p$  in the direction of vector  $\vec{pv}$ . Now, let  $e = uv$  be an edge of  $T$ . The vase of  $p$  and  $e$ , denoted  $\uparrow_p^e$ , is the region bounded by  $e$ ,  $\uparrow_p^u$ , and  $\uparrow_p^v$  (see Fig. 4).



■ **Figure 4** A vase.

■ **Figure 5** The situation in the proof of Lemma 3.

Vertices of the viewshed of  $p$  can have three types [11]. A vertex of type 1 is a vertex of  $T$ , of which there are clearly only  $n$ . A vertex of type 2 is the intersection of an edge of  $T$  and a vase. A vertex of type 3 is the intersection of a triangle of  $T$  and two vases.

► **Lemma 2.** *There are at most  $O(n \cdot \pi(T))$  vertices of type 2.*

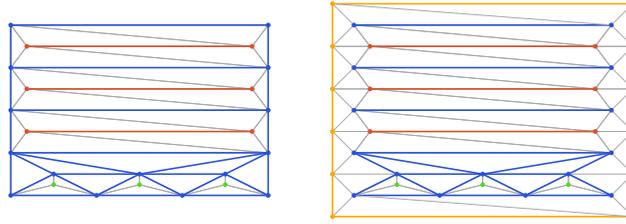
**Proof.** Consider an edge  $e$  of  $T$  and let  $H$  be the plane spanned by  $e$  and  $p$ . Consider the viewshed of  $p$  on  $e$ . Let  $qr$  be a maximal invisible portion of  $e$  surrounded by two visible ones. Since  $q$  and  $r$  are vertices of type 2, the open segments  $pq$  and  $pr$  pass through a point of  $T$ . On the other hand, for any point  $x$  in the open segment  $qr$ , there exist points of  $T$  above the segment  $px$ . This implies that there is a continuous portion of  $T$  above  $H$  such that the vertical projection onto  $H$  of this portion lies on the triangle  $pqr$ . Such portion has a local maximum in the direction perpendicular to  $H$  which is a convex and internal vertex of  $T$ . In consequence, each invisible portion of  $e$  surrounded by two visible ones can be assigned to a distinct point of  $T$  that is a local maximum in the direction perpendicular to  $H$ . Hence, in the viewshed of  $p$ ,  $e$  is partitioned into at most  $2\pi(T) + 3$  parts. ◀

► **Lemma 3.** *There are at most  $O(n \cdot \pi(T))$  vertices of type 3.*

**Proof.** Let  $q$  be a vertex of type 3 in the viewshed of  $p$ . Point  $q$  is the intersection between a triangle  $t$  of  $T$  and two vases, say,  $\uparrow_p^{e_1}$  and  $\uparrow_p^{e_2}$ ; see Fig. 5. Let  $r$  be the ray with origin at  $p$  and passing through  $q$ . Ray  $r$  intersects edges  $e_1$  and  $e_2$ . First, we suppose that  $e_1$  and  $e_2$  do not share any vertices and, without loss of generality, we assume that  $r \cap e_1$  is closer to  $p$  than  $r \cap e_2$ . Notice that  $r \cap e_2$  is a vertex of type 2 because it is the intersection of  $e_2$  and  $\uparrow_p^{e_1}$ , and  $\uparrow_p^{e_1}$  partitions  $e_2$  into a visible and an invisible portion. Thus, we charge  $q$  to  $r \cap e_2$ . If another vertex of type 3 was charged to  $r \cap e_2$ , then such a vertex would also lie on  $r$ . However, no point on  $r$  after  $q$  is visible from  $p$  because the visibility is blocked by  $t$ . Hence, no other vertex of type 3 is charged to  $r \cap e_2$ .

If  $e_1$  and  $e_2$  are both incident to a vertex  $v$ , since  $t \cap \uparrow_p^{e_1} \cap \uparrow_p^{e_2}$  is a type 3 vertex, we have that  $r$  passes through  $v$ . Therefore,  $q$  is the first intersection point between  $r$  (which can be seen as the ray with origin at  $p$  and passing through  $v$ ) and the interior of some triangle in  $T$ . Therefore, any vertex  $v$  of  $T$  creates at most a unique vertex of type 3 in this way. ◀

► **Theorem 4.** *Let  $T$  be a TIN terrain with  $n$  vertices and prickliness  $\pi(T)$ . The complexity of a viewshed is  $O(n \cdot \pi(T))$ . This bound is asymptotically tight.*



■ **Figure 6** Schematic top-down view of the classic quadratic construction (left), and the same adapted to have small prickliness (right). If an observer is at  $(0, -\infty)$ , the resulting view has quadratic complexity (see Fig. 1). Blue vertices/edges are low, red are medium height, and green are high. The right construction introduces a new height (yellow) between medium and high, and changes the triangulation slightly, to ensure that all convex vertices in the construction are green.

**Proof.** The upper bound on the complexity follows directly from Lemmas 2 and 3. Hence, we focus on the lower bound; i.e. we show that there exists a terrain  $T$  with prickliness  $\pi(T)$ , and a viewpoint on  $T$  with viewshed whose complexity is  $\Omega(n \cdot \pi(T))$ .

Consider the standard quadratic viewshed construction, composed of a set of front mountains and back triangles (Fig. 6 (left)). Notice that there can be at most  $\pi(T)$  mountains “at the front”. We add a surrounding box around the construction, see Fig. 6 (right), such that each vertex of the back triangles is connected to at least one vertex on this box. We set the elevation of the box so that it is higher than all the vertices of the back triangles, but lower than those of the front mountains. In this way, no vertex of the back triangles will be a local maximum in any direction, and all local maxima will come from the front. ◀

## 4 Prickliness computation

### 4.1 Algorithm for TINs

For every convex terrain vertex  $v$ , we compute the region of the unit sphere  $\mathbb{S}^2$  containing all vectors  $\vec{w}$  such that  $v$  is a local maximum of  $T$  in direction  $\vec{w}$ . As we will see, such a region is a cone and we denote it by  $co(v)$ . To compute  $co(v)$ , we consider all edges of  $T$  incident to  $v$ . Let  $e = vu$  be such an edge, and consider the plane orthogonal to  $e$  through  $v$ . Let  $H$  be the half-space which is bounded by this plane and does not contain  $u$ . We translate  $H$  so that the plane bounding it contains the origin; let  $H_e$  be the intersection of the obtained half-space with the unit sphere  $\mathbb{S}^2$ . The following property holds: For any unit vector  $\vec{w}$  in  $H_e$ , the edge  $e$  does not extend further than  $v$  in direction  $\vec{w}$ . We repeat this procedure for all edges incident to  $v$ , and consider the intersection  $co(v)$  of all the obtained half-spheres  $H_e$ . For any unit vector  $\vec{w}$  in  $co(v)$ , none of the edges incident to  $v$  extends further than  $v$  in direction  $\vec{w}$ . Since  $v$  is convex, this implies that  $v$  is a local maximum in direction  $\vec{w}$ .

Once we know all regions of type  $co(v)$ , computing the prickliness of  $T$  reduces to finding a unit vector that lies in the maximum number of such regions. To simplify, rather than considering these cones on the sphere, we extend them until they intersect the boundary of a unit cube  $\mathbb{Q}$  centered at the origin. The conic regions of type  $co(v)$  intersect the faces of  $\mathbb{Q}$  forming (overlapping) convex regions. Notice that the problem of finding a unit vector that lies in the maximum number of regions of type  $co(v)$  on  $\mathbb{S}$  is equivalent to the problem of finding a point on the surface of  $\mathbb{Q}$  that lies in the maximum number of “extended” regions of type  $co(v)$ . The second problem can be solved by computing the maximum overlap of convex regions using a topological sweep [6], for each face of the cube.

Computing the intersection between the extended regions of type  $co(v)$  (for all convex vertices  $v$ ) and the boundary of  $\mathbb{Q}$  takes  $O(n \log n)$  time, and topological sweep to find the maximum overlap takes  $O(n^2)$  time. We obtain the following:

► **Theorem 5.** *The prickliness of an  $n$ -vertex TIN terrain can be computed in  $O(n^2)$  time.*

## 4.2 Algorithm for DEMs

The prickliness of a DEM terrain can be computed using the same algorithm as for TINs: Each cell center can be seen as a vertex  $v$ , and its neighbors are the cell centers of its eight neighboring cells. The edges connecting  $v$  to its neighbors can then be used to compute  $co(v)$  as in Section 4.1, and the rest of the algorithm follows. However, DEM terrains have significantly more vertices, and vertices have on average more neighbors; this causes a significant increase in computation time and, more importantly, in memory usage. For this reason, in our experiments, the prickliness values for the DEM terrains were approximated.

The approximated algorithm discretizes the set of vectors that are candidates to achieve prickliness as follows: For every interior cell  $g$  of the terrain, we translate a horizontal grid  $G$  of size  $n$  by  $n$  and cell size  $s$  above the cell center  $v$  of  $g$  in a way that  $v$  and the center of  $G$  are vertically aligned and at distance one. Then the vectors considered as potential prickliness are those with origin at  $v$  and endpoint at some cell center  $c$  in  $G$ . For any such vector lying inside  $co(v)$ , the value of  $c$  gets incremented by one. When all interior cells of the terrain have been processed, a cell center of  $G$  with maximum value gives the approximated prickliness of the terrain. Cell size  $s$  was set to 0.05, based on the spread of the results on TIN terrains. This method should, in practice, produce a close approximation of prickliness.

## 5 Experiments

In this section we present our experimental setup. Our goals are to

- verify our hypothesis that existing topographic attributes do not provide a good indicator of the viewshed complexity,
- evaluate whether prickliness *does* provide a good indicator of viewshed complexity in practice, and
- evaluate whether these results are consistent for DEMs and TINs.

Furthermore, since in many applications we care about the visibility of multiple viewpoints (e.g. placing guards or watchtowers), we also investigate these questions with respect to the complexity of the common viewshed of a set of viewpoints. In this setting a point is part of the (common) viewshed if and only if it can be seen by at least one viewpoint. Note that since Theorem 4 *proves* that the complexity of a viewshed is proportional to the prickliness, our second goal is mainly to evaluate the practicality of prickliness. That is, to establish if this relation is also observable in practice or that the hidden constants in the big-O notation are sufficiently large that the relation is visible only for very large terrains.

Next, we briefly describe our implementations of the topographic attributes. We then outline some basic information about the terrain data that we use as input, and we describe how we select the viewpoints for which we compute the viewsheds.

**Implementations.** We consider prickliness and the topographic attributes from Section 2. To compute the prickliness we implemented the algorithm from Theorem 4 in C++ using CGAL 5.0.2 [23] and its *2D arrangements* [24] library.<sup>③</sup> We also implemented the algorithms

<sup>③</sup> Source code available from <https://github.com/GTMeijer/Prickliness>.

for TRI, TSI, and FD on TINs. These are mostly straightforward. To compute the viewsheds on TINs we implemented the hidden-surface elimination algorithm of [10] using CGAL.<sup>④</sup> We remark that our implementation computes the exact TIN viewsheds, as opposed to previous studies that only considered fully visible triangles (e.g., [19]).

To compute the prickliness on DEMs we used the algorithm from Section 4.2. For TRI, TSI, and FD on DEMs we used the implementations available in ArcGIS Pro 2.5.1 [7]. To compute viewsheds on DEMs we used the builtin tool “Viewshed 2” in ArcGIS, which produces a raster with boolean values that indicate if a cell is visible or not. To get a measure of complexity similar to that of the TINs we use the “Raster to Polygon” functionality of ArcGIS (with its default settings) to convert the set of TRUE cells into a set of planar polygons (possibly with holes). We use the total number of vertices of these polygons as the complexity of the viewshed on a DEM.

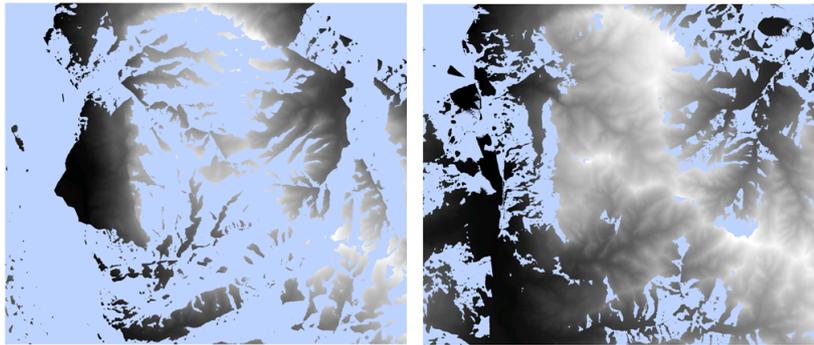
**Terrains.** We considered a collection of 52 real-world terrains around the world. These terrains were handpicked in order to cover a large variety of landscapes, with varying ruggedness, including mountainous regions (Rocky mountains, Himalaya), flat areas (farmlands in the Netherlands), and rolling hills (Sahara), and different complexity. We obtained the terrains through the *Terrain* world elevation layer [8] in ArcGIS [7]. Each terrain is represented as a 10-meter resolution DEM of size  $1400 \times 1200$ . According to past studies the chosen resolution of 10 meters provides the best compromise between high resolution and processing time of measurements [15, 25]. The complete list of terrains with their extents can be found in [17].

We generated a TIN terrain for each DEM using the “*Raster to TIN*” function in ArcGIS [7]. This function generates a Delaunay triangulation to avoid long, thin triangles as much as possible. With the *z-tolerance* setting, the triangulation complexity can be controlled by determining an allowed deviation from the DEM elevation values. We considered TIN terrains generated using a *z-tolerance* of 50 meters. This resulted in TINs where the number of vertices varied between 30 and 5808 (with an average of 1547 vertices). Their distribution can be seen in Fig. 8.

**Viewpoints.** Kim et al. [13] found that placing the viewpoints at peaks typically produces viewsheds that cover hilltops, but not many valleys, whereas placing viewpoints in pits typically covers valleys but not hilltops. This leads us to consider three different strategies to pick the locations of the viewpoints: picking “high” points (to cover peaks), picking “low” points (to cover valleys), and picking viewpoints uniformly at random. To avoid clusters of high or low viewpoints we overlay an evenly spaced grid on the terrain, and pick one viewpoint from every grid cell (either the highest, lowest, or a random one). We pick these points based on the DEM representation of the terrains, and place the actual viewpoints one meter above the terrain to avoid degeneracies. We use the same locations in the TINs in order to compare the results between TINs and DEMs (we do recompute the *z*-coordinates of these points so that they remain 1m above the surface of the TIN). The resulting viewsheds follow the expected pattern; refer to Fig. 7. In our experiments, we consider both the complexity of a viewshed of a single viewpoint as well as the combined complexity of a viewshed of nine viewpoints (picked from a  $3 \times 3$  overlay grid). Results of Kammer et al. [12] suggest that for the size of terrains considered these viewpoints already cover a significant portion of the terrain, and hence picking even more viewpoints is not likely to be informative.

---

<sup>④</sup> Source code available from [https://github.com/GTMeijer/TIN\\_Viewsheds](https://github.com/GTMeijer/TIN_Viewsheds).



■ **Figure 7** (left) A joint viewshed (blue) created from viewpoints placed on the highest points. (right) A joint viewshed (blue) created from viewpoints placed on the lowest points.

■ **Table 1** The correlation coefficients ( $R$  values) between the attributes and viewshed complexity.

	TIN						DEM					
	highest		lowest		random		highest		lowest		random	
	single	multi	single	multi	single	multi	single	multi	single	multi	single	multi
Prick	0.75	0.97	0.41	0.83	0.64	0.93	0.63	0.90	0.10	0.19	0.18	0.64
TRI	0.44	0.58	0.62	0.72	0.61	0.63	-0.52	-0.38	-0.27	-0.30	-0.34	-0.43
TSI	0.45	0.69	0.69	0.81	0.63	0.75	-0.53	-0.40	-0.24	-0.23	-0.32	-0.42
FD	-0.56	-0.73	-0.64	-0.78	-0.66	-0.76	0.11	-0.27	0.27	0.25	0.27	-0.01

**Analysis.** For each topographic attribute we consider its value in relation to the complexity of the viewsheds. In addition, we test if there is a correlation between the viewshed complexity and the attribute in question. We compute their sample correlation coefficient (Pearson correlation coefficient)  $R$  to measure their (linear) correlation. The resulting value is in the interval  $[-1, 1]$ , where a value of 1 implies that a linear increase in the attribute value corresponds to a linear increase in the viewshed complexity. A value of  $-1$  would indicate that a linear increase in the attribute leads to a linear decrease in viewshed complexity, and values close to zero indicate that there is no linear correlation.

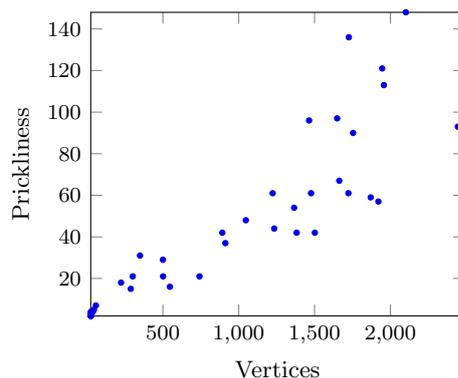
## 6 Results

We start by investigating the prickliness values compared to the complexity of the terrains considered. These results are shown in Fig. 8. We can see that the prickliness is generally much smaller than the number of vertices in the (TIN representation of the) terrain. In Fig. 9 we also see one the  $\pi_{\bar{v}}$  values for orientation vectors near  $(0, 0, 1)$  (recall that the maximum over all orientations defines the prickliness).

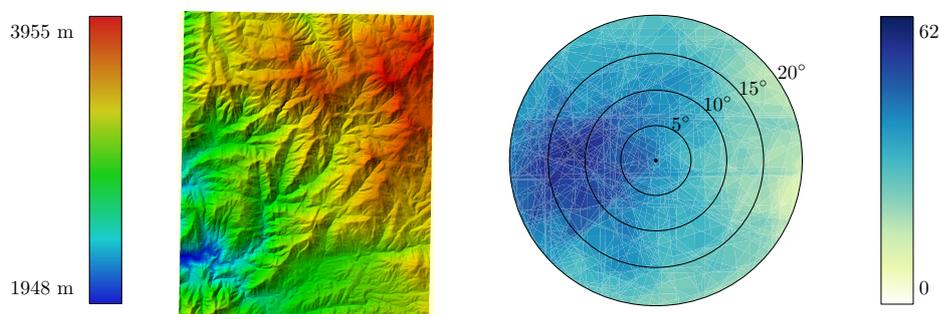
### 6.1 Topographic attributes and viewshed complexity

For each of the terrains we compute the viewshed of one or nine viewpoints, for three viewpoint placement strategies, and analyze the complexity of the viewshed as a function of the topographic attributes both for TINs and DEMs. Table 1 summarizes the correlation between the attributes and the viewshed complexity for each case.

## 10:12 Terrain Prickliness: Theoretical Grounds for High Complexity Viewsheds



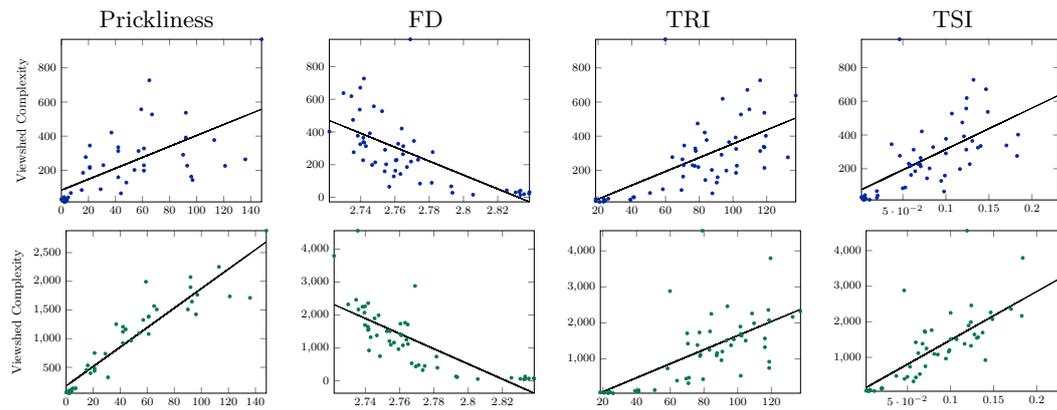
■ **Figure 8** The prickliness values for the terrains we considered.



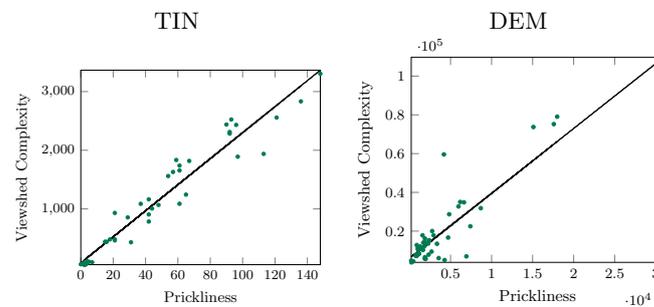
■ **Figure 9** (left) A real-world terrain with 583 vertices from the neighborhood of California Hot Springs whose prickliness is only 62. (right) The value  $\pi_{\vec{v}}$  for vectors near (0, 0, 1).

**Single viewpoint.** The first row in Fig. 10 shows the full results for a randomly placed viewpoint on the TIN. Somewhat surprisingly, we see that terrains with high fractal dimension have a low viewshed complexity. For the other measures, higher values tend to correspond to higher viewshed complexities. However, the scatter plots for these two measures show a large variation. The scatter plots for the other placement strategies (highest and lowest) look somewhat similar, hence the strategy with which we select the viewpoints does not seem to have much influence in this case. None of the four attributes shows a strong correlation in this case (see also Table 1). Prickliness shows weak-medium correlation in three out of six cases, strong correlation for one case – viewpoints at highest points – and no correlation for two cases with viewpoints at lowest points. The other attributes show an even weaker correlation in general.

**Multiple viewpoints.** The results comparing the topographic attributes to the complexity of the common viewshed of multiple viewpoints, selected from a  $3 \times 3$  overlay grid (refer to Section 5) can be found in the second row of Fig. 10. Again, fractal dimension shows an inverse behavior. In contrast, the other three attributes show now a much clearer positive correlation with viewshed complexity. In this case the prickliness shows the strongest correlation in all but one case (that of viewpoints at lowest points). In particular when placing the viewpoints on highest points within the overlay grids the correlation is strong. See also Fig. 11 (left).



■ **Figure 10** The viewshed complexity on a TIN. First row: single random viewpoint. Second row: common viewshed of multiple (nine, selected from a  $3 \times 3$  overlay) random viewpoints.



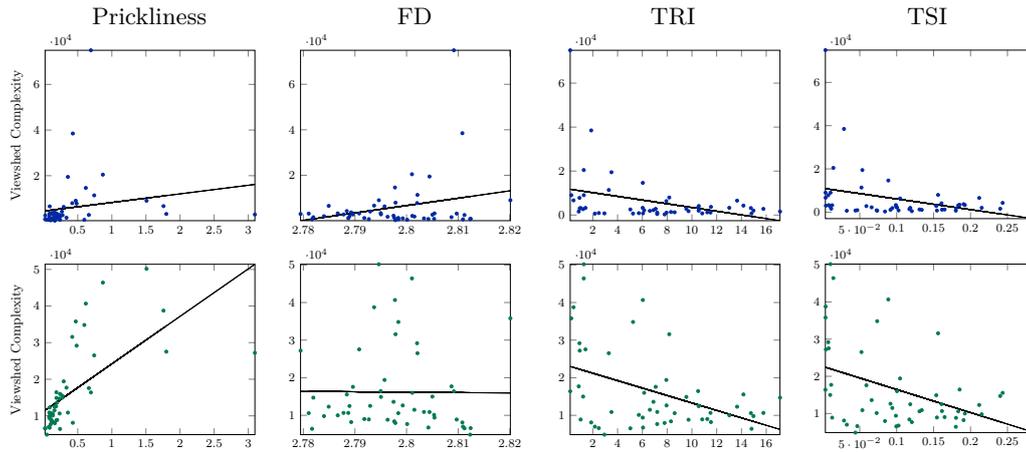
■ **Figure 11** The complexity of the common viewshed of nine viewpoints placed using the highest strategy on a TIN (left) or on a DEM (right).

**DEM results.** The scatter plots for randomly placed viewpoints can be found in Fig. 12. The first row again corresponds to the complexity of a single viewshed, whereas the second row corresponds to the complexity of the common viewshed of nine viewpoints selected from a  $3 \times 3$  overlay. In contrast to TINs, for DEMs all measures show no to weak correlation values. The correlation between prickliness and the viewshed complexity is the highest with a positive correlation value of 0.629. However, the scatter plot (Fig. 11 right) suggests that this value is not very meaningful.

Something similar occurs in the case of viewsheds of DEMs for multiple viewpoints, where none of the measures seems to have a statistical significance, with most of the  $R^2$  values being below 0.200. The exception to this is the correlation between prickliness for viewpoints on highest points, which shows a very strong relationship.

## 7 Discussion

The experimental results for TINs confirm our hypotheses. We can see a clear correlation between the viewshed complexity and the prickliness, especially when multiple viewpoints are placed on the highest points. In contrast, this is not evident for the other three topographic attributes considered. The terrain ruggedness index (TRI) and terrain shape index (TSI) show some very weak positive correlation, but not as strong as prickliness. This could be explained by the fact that TRI and TSI only consider a fixed neighborhood around each



■ **Figure 12** The viewshed complexity on a DEM. First row: single random viewpoint. Second row: common viewshed of multiple (nine, selected from a  $3 \times 3$  overlay) random viewpoints. The prickliness values are times  $10^4$ .

point, making them local measures unable to capture the whole viewshed complexity. Indeed, a small (local) obstruction can be enough to significantly alter the value for any of these attributes. The fractal dimension (FD) seems to be even worse at predicting viewshed complexity. Unlike TRI and TSI, this topographic attribute considers the variability within an area of the terrain as opposed to a fixed-radius neighborhood. Taking a closer look at the FD values for both terrain datasets shows a minimal variation, with most of them being close to 3.0, which, according to Taud et al. [22], indicates a nearly-constant terrain. These results seem to indicate that this measure fails to detect the variation in elevation levels with the chosen parameters.

The situation for DEM terrains is less clear. Only for viewsheds originating from the highest points we see a strong correlation between prickliness and viewshed complexity. When the viewpoints are placed at the lowest points of the DEM terrains, the correlation disappears. Since the prickliness measures the amount of peaks in the terrain in all possible (positive) directions, this means that when a viewpoint is placed at the highest elevation and the viewshed gets split up by the protrusions (which seem to be accurately tracked by prickliness), there is a strong correlation. However, when the viewpoints are placed at the lowest points, the viewsheds become severely limited by the topography of the terrain surrounding them. Even when placing multiple viewpoints, these viewsheds do not seem to encounter enough of the protrusions that are detected by the prickliness measure for viewpoints placed at high points.

One possible explanation for the difference between the results on the TIN and DEM terrains for prickliness could be attributed to the difference in resolution between the DEMs and TINs used. The DEMs used consisted of 1.68M cells of 10m size, while the TINs – generated with an error tolerance of 50m – had 1547 vertices on average. While it would have been interesting to use a higher resolution TIN, this was not possible due to the high memory usage of the prickliness algorithm. Another possible explanation for the mismatch between the results for TINs and DEMs may be on the actual definition of prickliness, which is more natural for TINs than for DEMs. Indeed, it can be seen in the results that the prickliness values for DEMs are much higher than for TINs, which could indicate that the definition is too sensitive to small terrain irregularities.

## 8 Conclusion

We established that prickliness is a reasonable measure of potentially high viewshed complexity, at least for TINs, confirming our theoretical results. Moreover, prickliness shows a much clearer correlation with viewshed complexity than the three other terrain attributes considered.

One aspect worth further investigation is its correlation for DEMs, which seems to be much weaker. One explanation for this might be that the definition of prickliness is more natural for TINs than for DEMs, but there are several other possible explanations, and it would be interesting future work to delve further into this phenomenon. Having established that prickliness can be a useful terrain attribute, it remains to improve its computation time, so it can be applied to larger terrains in practice.

Finally, during our work we noticed that several of the terrain attributes are defined locally, and are parameterized by some neighborhood size. Following previous work, we aggregated these local measures into a global measure by averaging the measurements. It may be worthwhile to investigate different aggregation methods as well. This also leads to a more general open question on how to “best” transform a local terrain measurement into a global one.

---

## References

- 1 Brent C. Chamberlain and Michael J. Meitner. A route-based visibility analysis for landscape management. *Landscape and Urban Planning*, 111:13–24, 2013.
- 2 Maria Danese, Gabriele Nolè, and Beniamino Murgante. Identifying viewshed: New approaches to visual impact assessment. In *Geocomputation, Sustainability and Environmental Planning*, pages 73–89. Springer, 2011.
- 3 Mark de Berg, Herman Haverkort, and Constantinos P. Tsirogiannis. Visibility maps of realistic terrains have linear smoothed complexity. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, pages 163–168. ACM, 2009.
- 4 D J Dean. Improving the accuracy of forest viewsheds using triangulated networks and the visual permeability method. *Canadian Journal of Forest Research*, 27(7):969–977, 1997.
- 5 Youfu Dong, Guoan Tang, and Ting Zhang. a Systematic Classification Research of Topographic Descriptive Attribute in Digital Terrain Analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 B2:357–362, 2008.
- 6 Herbert Edelsbrunner and Leonidas J. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38(1):165–194, 1989.
- 7 Environmental Systems Research Institute (ESRI). Arcgis pro (2.5.1), May 2020.
- 8 Environmental Systems Research Institute (ESRI). Terrain, scale: 10m, February 2020.
- 9 W. Randolph Franklin and Christian Vogt. Multiple observer siting on terrain with intervisibility or lo-res data. In *XXth Congress, International Society for Photogrammetry and Remote Sensing*, pages 12–23, 2004.
- 10 Michael T. Goodrich. A polygonal approach to hidden-line and hidden-surface elimination. *CVGIP: Graphical Models and Image Processing*, 54(1):1–12, January 1992.
- 11 Ferran Hurtado, Maarten Löffler, Inês Matos, Vera Sacristán, Maria Saumell, Rodrigo I. Silveira, and Frank Staals. Terrain visibility with multiple viewpoints. *International Journal of Computational Geometry & Applications*, 24(04):275–306, 2014.
- 12 Frank Kammer, Maarten Löffler, Paul Mutser, and Frank Staals. Practical approaches to partially guarding a polyhedral terrain. In *Proc. 8th International Conference on Geographic Information Science*, LNCS 8728, pages 318–332, 2014.

## 10:16 Terrain Prickliness: Theoretical Grounds for High Complexity Viewsheds

- 13 Young-Hoon Kim, Sanjay Rana, and Steve Wise. Exploring multiple viewshed analysis using terrain features and optimisation techniques. *Computers & Geosciences*, 30(9):1019–1032, 2004.
- 14 Benoit B Mandelbrot. *The fractal geometry of nature*. W.H. Freeman, New York, 1982.
- 15 J. J. Maynard and M. G. Johnson. Scale-dependency of LiDAR derived terrain attributes in quantitative soil-landscape modeling: Effects of grid resolution vs. neighborhood extent. *Geoderma*, 230-231:29–40, 2014.
- 16 W. Henry McNab. Terrain shape index: Quantifying effect of minor landforms on tree height. *Forest Science*, 35:91–104, 1989.
- 17 Gert Meijer. Realistic terrain features and the complexity of joint viewsheds. Master’s thesis, Utrecht University, 2020.
- 18 Esther Moet, Marc van Kreveld, and A. Frank van der Stappen. On realistic terrains. *Computational Geometry*, 41(1):48–67, 2008.
- 19 Philip D Riggs and Denis J Dean. An investigation into the causes of errors and inconsistencies in predicted viewsheds. *Transactions in GIS*, 11(2):175–196, 2007.
- 20 Shawn J. Riley, Stephen D. DeGloria, and Robert Elliot. A terrain ruggedness index that quantifies topographic heterogeneity. *Journal of Science*, 5:23–27, 1999.
- 21 Uta Schirpke, Erich Tasser, and Ulrike Tappeiner. Predicting scenic beauty of mountain regions. *Landscape Urban Plan.*, 111:1–12, 2013.
- 22 Hind Taud and Jean-François Parrot. Measurement of DEM roughness using the local fractal dimension. *Géomorphologie : relief, processus, environnement*, 11(4):327–338, 2005.
- 23 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.0.2 edition, 2020.
- 24 Ron Wein, Eric Berberich, Efi Fogel, Dan Halperin, Michael Hemmer, Oren Salzman, and Baruch Zukerman. 2d arrangements. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.0.2 edition, 2020.
- 25 Weihua Zhang and David R Montgomery. Digital elevation model grid size, landscape representation, and hydrologic simulations. *Water Resources Research*, 30(4):1019–1028, 1994.

# User Preferences and the Shortest Path

Isabella Kreller ✉

University of Regensburg, Germany

Bernd Ludwig ✉ 

University of Regensburg, Germany

---

## Abstract

Indoor navigation systems leverage shortest path algorithms to calculate routes. In order to define the “shortest path”, a cost function has to be specified based on theories and heuristics in the application domain. For the domain of indoor routing, we survey theories and criteria identified in the literature as essential for human path planning. We derive quantitative definitions and integrate them into a cost function that weights each of the criteria separately. We then apply an exhaustive grid search to find weights that lead to an ideal cost function. “Ideal” here is defined as guiding the algorithm to plan routes that are most similar to those chosen by humans. To explore which criteria should be taken into account in an improved pathfinding algorithm, eleven different factors whose favorable impact on route selection has been established in past research were considered. Each factor was included separately in the Dijkstra algorithm and the similarity of thus calculated routes to the actual routes chosen by students at the University of Regensburg was determined. This allows for a quantitative assessment of the factors’ impact and further constitutes a way to directly compare them. A reduction of the number of turns, streets, revolving doors, entryways, elevators as well as the combination of the aforementioned factors was found to have a positive effect and generate paths that were favored over the shortest path. Turns and the combination of criteria turned out to be most impactful.

**2012 ACM Subject Classification** Human-centered computing; Human-centered computing → Empirical studies in ubiquitous and mobile computing

**Keywords and phrases** Pedestrian Navigation Systems, Wayfinding, Computation of Optimal Paths, User Preferences for Best Routes

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.11

## 1 Introduction

Finding a way through their immediate environment to reach a sought destination is a task highly relevant to humans since they see themselves confronted with it regularly. Due to the available options of traveling and visiting unfamiliar cities in foreign countries, the cultural differences as well as the possible language barrier may further complicate wayfinding tasks. Being able to facilitate situations like these, location-based navigation services, particularly those intended for car navigation, enjoy great popularity and have established themselves as indispensable supporting tools for navigation in the modern world, as indicated by the high usage of Google Maps [17].

Although still overshadowed by the ubiquity of outdoor navigation systems, indoor wayfinding is not anymore an obscure area of research and it might be that in this setting navigational support might be needed most. Wayfinding, meaning the goal-oriented process of a person finding a predefined destination [2], often involving navigation through large areas without initial perception of the target location [27], is deemed a cognitive challenging task requiring a variety of mental resources [35].

In addition, indoor spaces exhibit structural attributes, such as the additional dimensionality inherent in multi-storied buildings [18] or the lack of reliable methods for localization [42], which give rise to further difficulties and require current systems to be adapted to this navigational context.



© Isabella Kreller and Bernd Ludwig;

licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Versteegen; Article No. 11; pp. 11:1–11:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 11:2 User Preferences and the Shortest Path

This likewise concerns the pathfinding algorithm: due to the current lack of geospatial research regarding this setting, employed algorithms have shown little to no deviation from optimizing only in regard to the route's efficiency disregarding human preference. Previous studies, as discussed in subsections 2.2, 2.3, and 3.3, have determined a multitude of diverse criteria as influential to people's path choices, yet quantification of their impact and applicability in indoor environments remain to be studied.

### **2** Related Work

#### 2.1 Difficulties of Indoor Navigation

The reasons for which indoor navigation systems lack the popularity and widespread use their outdoor counterparts are enjoying are twofold.

Firstly, there exist additional difficulties regarding the implementation of indoor navigation systems. The additional third dimension, the height, inherent in multi-level buildings creates challenges regarding a suitable design for the visual projection onto the small two-dimensional screens of smartphones [5]. This is important for providing easily comprehensible and unambiguous instructions to the user to direct them to the correct floor. Hence, conventional visualizations and user interfaces of map-based applications are not predestined for wayfinding support inside buildings.

Yet the most crucial obstacle to overcome is likely the absence of a reliable method of localization. Whereas the vast majority of outdoor navigation systems make use of Global Positioning System (GPS) signals [42], this does not constitute a viable option indoors. These signals manage to pass through the walls and ceilings of buildings only sporadically, rendering it an unsatisfactory solution for optimal navigational support [42]. Although numerous alternatives, including WiFi, Bluetooth, and Infrared, have been proposed, these too entail drawbacks regarding performance, cost, or reliability. [42]

Furthermore, human orientation may also prove more challenging in this setting. The third dimension, i.e. multiple stories inside buildings, creates challenges not only for the system's ability to display information but for human wayfinding as well. Staircases have been identified as the location where mistakes are being made most frequently during a wayfinding task [18]; other research supports the idea that people struggle with vertical, rather than horizontal, movement, which might be explained by the mistaken assumption that a building's layout is identical across floors [43]. A restricted field of vision due to walls and corners makes the usage of landmarks as points of reference only possible if they are local or in direct line of view, whereas in outdoor spaces global landmarks, such as towers or the city skyline, can also support wayfinding. [52].

#### 2.2 Influential Factors

There is reason to contest and doubt that humans' behavior is purely rational and can be explained with conscious and deliberate reasoning alone for we are susceptible to numerous biases in our ways of thinking and decision making [23].

The same holds for wayfinding behavior: while the most logical path to the desired destination would generally be the shortest or least time consuming one, this does not align with actual human conduct or preference [3, 14, 3, 54].

Afyouni, Ray, and Claramunt [1] suggest a classification of the context in which decision making takes place that can also be applied to wayfinding tasks. The user-centered context takes into account the navigator's profile, such as mental capabilities of visualization [44],

or rotation of objects [32], their age [45] and gender [28], or cultural background [20]. The context of execution, on the other hand, describes qualities of the information systems which aid the user; the representation of maps and real-world references used by mobile applications [38] has been found to impact navigational success, and so do conventional means of support, such as signs and paper maps to varying degrees.

What is most relevant for the adjustment of the pathfinding algorithm, however, can be described by the environmental context.

According to Lynch [29], the components most crucial for the formation of mental maps are paths, edges, districts, and landmarks, which, belonging to the same mental models, exist in connection and mutual interference. This concept fits well into the research conducted, which suggests that humans mentally visualize their environment as being structurally equivalent to a network of streets and paths [13, 27, 33].

In this sense, the definition of a path's complexity contrived by Golledge [14] and since adopted into the work of many others [11, 19, 40] is to consider the number of streets or turns connecting them.

Alternative approaches [4, 12, 16, 40, 51] of achieving simplicity take into account the number of decision points, i.e. intersections of multiple streets, or their branching factor. It is defined by the number of streets meeting, thus considered a measure of difficulty or complexity of a decision point [37] and increasing the possibility of losing one's way [2].

Rather than the number of branches, Dalton [8] found a preference for choosing the maximum, rather than mean or minimum, angles of paths available at decision points, meaning that people were more likely to choose the way that was aligned most with from where they were coming, presumably as a complexity-minimizing strategy. Further observations support the idea of people's predilection towards maintaining linearity in their routes [7, 34, 41, 53]. Similarly, another study points towards a human preference of decreasing the angular distance to the destination (Turner, 2009) at decision points, favoring this over a reduction of the Manhattan distance.

Other than a preference for simple paths, the absence or presence of certain path entities could serve as a decisive factor. Landmarks, defined by their conspicuity and recognizability [6, 29] commonly function as points of reference and may help reduce cognitive load for orientation [15]. For staircases, both a preference [24] and avoidance [24, 26] have been identified.

### 2.3 Alternatives to the Shortest Path

Variants for the cost function in path planning have been proposed with different objectives in mind. Grum [16] describes a way of extending the Dijkstra algorithm so that it incorporates a weight for the risk of making a wrong decision at an intersection proportional to the distance of the mistakenly chosen path, assuming that the navigator notices their mistake at the next decision point and retraces their way. In this manner, not the shortest path, but the one with the least danger of getting lost is computed. Yet, the question remains open how the risk for wrong decisions may be estimated for arbitrary environments.

Duckham and Kulik [11] proposed a way of calculating the simplest path, using solely a measure for ease of navigation by weighting route instructions without taking distance into account at all. Instructions needed to navigate decision points were penalized according to adjusted slot values for turns and branches as proposed by Mark [31], yielding the sought-after advantages.

## 11:4 User Preferences and the Shortest Path

Physically constrained people constitute a group of users which might benefit immensely from adjustments to conventional navigation systems. A proposed way of achieving this is by taking into account edge weights other than distances, thus considering specific user preferences such as avoiding stairs or minimizing turns which can be input directly by the user [30].

In contrast, introducing a measure of beauty as a way of suggesting short routes that are perceived to be emotionally pleasant was accomplished by crowdsourcing of people's judgments of places along the dimensions beautiful, quiet, and happy, making use of pictures from Google Street View or taken by volunteers [39]. Such an aesthetic criterion has not been considered by the persons who provided us with their preferred routes, because they tried to reach their destination as quickly and easily as possible. However, [22] report that minimizing the number of turns is a key criterion for route planning in outdoor environments. Such routes seem to be more easily perceivable and cognitively less demanding.

In our study we evaluate, whether this result can be transferred to indoor environments and whether other criteria (e.g. level changes or use of elevators) that result from indoor architectural constraints also influence human decision making for route planning.

### 3 Route Calculation

To operationalize the above research question, we make use of the indoor navigation system URWalking [36, 49] ([urwalking.ur.de](http://urwalking.ur.de)). It implements DIJKSTRA's algorithm [10] to determine the shortest path in terms of the distance between starting point and destination in meters.

#### 3.1 User Preferred Routes

Our empirical study is based on a data set of 221 routes, collected both during the summer (100 paths) and winter semester (121 paths) at the University of Regensburg, thus accommodating for differences in weather and temperature. All participants, 129 of which were male, stated to be highly familiar with the environment and ranged in age from 19 to 33 (mean age = 22.93, SD = 2.67). As means for data collection, participants were asked to walk and describe their daily route on the campus. The university campus spans over an expanse of 1 km<sup>2</sup> and includes 16 buildings with a sum of 86 individual floors and 5228 rooms, encompassing both paths indoors and outdoors [36].

The aim of the data collection was to find out whether modifications of URWalking's path planning algorithm could lead to routes users enjoy taking, so as to find a compromise between the theoretically minimal walking distance and the distance of preferred paths (i.e. can we approximately simulate human decision making using DIJKSTRA's efficient algorithm?). For the envisioned modifications, we identified criteria in research literature that are argued to impact human decision making during wayfinding tasks.

#### 3.2 Criteria

The criteria that will be described in the following were chosen due to the extent of the research they are supported by and applied in, and also based on the feasibility of their incorporation into the DIJKSTRA algorithm. The scope of literature on the measurement of path complexity suggests that there is no agreed-upon answer as to how to define it. Although a few studies have suggested more elaborate ways of ruling how complex a path is, a few elementary classifications resurface frequently across research papers. We identified 11 factors:

A straightforward way to determine complexity is by considering the **(1) number of turns** a route contains. Regarding the definition of what constitutes a turn in navigation the answers once again are ambiguous. According to the Cambridge Dictionary [47], a turn can be considered “a change in the direction in which you are moving or facing”. Although in some studies a sufficient change in directions achieved only when the angle between two path segments is smaller or equal to 90 degrees [19, 50], others interpret every branch at a decision point to count as a turn [11, 25].

A related factor influential to wayfinding is the **(2) number of streets**, meaning the path segments separated by decision points or turns of any kind [19]. Considering only turns up to a maximum angle of 90 degrees would not account for the number of streets whereas disregarding the angle entirely would run the risk of accounting for turns that might not be perceived as such due to their small deviation from a linear path. For these reasons, both of the definitions of what constitutes a turn were examined: weighting turns regardless of their angle will acknowledge the number of streets more completely, and following the definition in accordance with [19] comprises a more traditional way of regarding turns and assessing their impact independently.

The **(3) number of decision points** themselves was considered for route calculation as well. Since the number of paths at a junction demanding a decision is two in addition to the path the navigator came from, a decision point in this study was defined as an intersection of three or more paths; their exact number was disregarded and every path located at a point which fits this definition was penalized in the same way.

The **(4) branching factor**, meaning the number of intersecting paths, was regarded separately. To weigh the paths proportionally to their quantity, every path at a decision point received the additional weight times the number of intersecting paths. These weights were taken into account only at places that qualify as decision points corresponding to the aforementioned definition.

The findings on which paths at intersections constitute the most likely choice are ambiguous, which is why two divergent findings were considered. One is the preference for selecting the path which most closely aligns with the global direction of the destination [48], being equal to the avoidance of all paths other than the one with the **(5) minimum deviation angle**.

Alternatively, a preference for **(6) linearity** [7] might influence the person navigating to continue their route along the most linear path, i.e. the path with the largest angle (between 0 and 180 degrees) in regard to the direction from which the navigator arrived. For this, all except the most linear branch at a decision point were penalized, allowing for the option of exempting two branches in case of identical largest angles. To keep with the definition of linearity as well as Dalton’s study [8], the condition was added that the largest angle had to span 150 degrees or more. If there was no such angle, e.g. in a T-crossing, no weighting was carried out.

Due to the contradictory nature of the literature regarding the preference for **(7) staircases** in routes, both a preference for and avoidance of stairs were considered. **(8) Elevators** constituted the only alternative for changing floors at the university; hence the avoidance of stairs, being equal to a preference for elevators, was implemented by adding weights for every staircase and vice versa.

Another element to be taken into consideration for indoor environments exclusively is doors. The path network used for calculating routes includes different types of nodes that differentiate between **(9) doorways**, **(10) entrances** to closed premises such as offices or lecture halls, termed entryways, and **(11) revolving doors**. Since it has been found that

## 11:6 User Preferences and the Shortest Path

doors take on the function of important landmarks in indoor spaces, and serve as transitions between separate spaces [46], the effect of doors, with the above-mentioned differentiation, will be included in the analysis.

We modified the original URWalking path finding algorithm such as to separately take the 11 above factors into account and to identify their optimal weights for calculating optimal routes. This constitutes a method of path calculation that, while still maintaining the objective of minimizing distance, can easily integrate different factors and assign them distinct levels of importance.

■ **Algorithm 1** DIJKSTRA's algorithm (source: Wikipedia).

---

**Result:** Shortest paths to all nodes starting from **source**

```
create vertex set Q;
for each vertex  $v$  in Graph do
    dist[ $v$ ] ← INFINITY;
    prev[ $v$ ] ← UNDEFINED;
    add  $v$  to Q;
    dist[source] ← 0;
end
while  $Q$  is not empty do
     $u$  ← vertex in  $Q$  with min dist[ $u$ ];
    remove  $u$  from  $Q$ ;
    for each neighbor  $v$  of  $u$  do
        alt ← dist[ $u$ ] + wlength( $u, v$ );
        if alt < dist[ $v$ ] then
            dist[ $v$ ] ← alt;
            prev[ $v$ ] ←  $u$ ;
        end
    end
end
return dist[], prev[];
```

---

The standard shortest path definition of  $wlength(x, y)$  is the metric distance  $length(x, y)$  between two nodes  $x$  and  $y$  connected by an edge. The operationalization of the identified factors is quite straightforward: we simply modified the cost function used by DIJKSTRA's algorithm (see Algorithm 1). The key to taking the factors into account is a modification of the function  $length(x, y)$  for the cost of edges in the graph. In the standard version, the function computes the distance between  $x$  and  $y$  in meters. In our modifications, we add artificial, heuristic costs to  $length(x, y)$ . We apply the idea that a factor can be accounted for by extending the distance between  $x$  and  $y$ , i.e. by making artificially the edge  $(x, y)$  more expensive. For each factor, a particular criterion determines whether it actually applies for an edge currently considered by DIJKSTRA's algorithm:

1. number of turns:

$$wlength(x, y, w) = \begin{cases} length(x, y) + w & \text{angle}(x, y) \leq 90 \\ length(x, y) & \text{otherwise} \end{cases}$$

2. number of streets:

$$wlength(x, y, w) = \begin{cases} length(x, y) + w & \text{angle}(x, y) \leq 180 \\ & \text{(route does not continue straight on after } y) \\ length(x, y) & \text{otherwise} \end{cases}$$

3. number of decision points:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & x \text{ has at least three neighbours} \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

4. branching factor:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + n \cdot \text{weight} & x \text{ has exactly } n \text{ neighbours and } n \geq 3 \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

5. minimum deviation angle:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & x \text{ has at least three neighbours} \\ & \text{and } y \neq \arg \min_z \text{angle}(x, z) \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

6. linearity:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & x \text{ has at least three neighbours} \\ & \text{and } y \neq \arg \max_z \text{angle}(x, z) \\ & \text{and } \max_z \text{angle}(x, z) \geq 150 \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

7. staircases:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & y \text{ is an edge of type } \mathbf{staircase} \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

8. elevators:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & y \text{ is an edge of type } \mathbf{elevator} \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

9. doorways:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & y \text{ is of type } \mathbf{doorway} \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

10. entrances:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & y \text{ is of type } \mathbf{entrance} \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

11. revolving doors:

$$\text{wlength}(x, y, w) = \begin{cases} \text{length}(x, y) + w & y \text{ is of type } \mathbf{revolving-door} \\ \text{length}(x, y) & \text{otherwise} \end{cases}$$

In these definition  $w$  is a constant whose value determines by how many meters the length of an edge between  $x$  and  $y$  is extended. Consequentially, the shortest path gets “longer” the more instances of the respective considered factor it includes.

With this model, our main task is now to determine values of  $w$  for each factor that maximizes the similarity between routes calculated by URWalking and those chosen by the participants of our study.

### 3.3 Systematic Search for Optimal Weights

Our intuition is that a meaningful assessment of *similarity between two routes*  $r_1$  and  $r_2$  reflects the length of shared path segments, i.e. edges in  $r_1$  and  $r_2$ :

$$\text{similarity}(r_1, r_2) = \frac{\sum_{e \in r_1 \wedge e \in r_2} \text{length}(e)}{\min(\text{length}(r_1), \text{length}(r_2))}$$

with  $\text{length}(r) = \sum_{e \in r} \text{length}(e)$  for any route  $r$  and  $\text{length}(e) = \text{length}(x, y)$  with  $x$  being the source node and  $y$  the sink node of any edge  $e$ .

In other words, the calculated Similarity Score  $\text{similarity}(r_1, r_2)$  was defined as the sum of the distances belonging to shared edges divided by the length of the shorter path (analogous to [21]). Following Delling et al.’s [9] approach, the mean of the similarity scores across all path comparisons was then considered for assessment of the algorithm, i.e. each factor’s individual impact on wayfinding.

## 11:8 User Preferences and the Shortest Path

To identify an optimal weight  $w$  for each factor, we performed a systematic grid search for this one-dimensional parameter. Overall, we performed 11 independent searches considering a different factor in each search.

In such a search, we first set  $w = 0$  to identify the shortest path in terms of the metric distance between a fixed start and destination. Then we iterated over  $w$  and incremented it by 1 in each iteration. As the cost function  $wlength(x, y, w)$  computes costs in meters, incrementing  $w$  by 1 adds one meter to the physical distance between  $x$  and  $y$  allowing a detour of 1 meter for a path still to be identified as the shortest path, this time however taking one of the identified factors into account. After having calculated this shortest path for a fixed  $w$  we computed the similarity score to preferred routes in our data set with identical start and destination resulting in a list of similarity scores for which we could easily calculate a mean similarity. We repeated this process for all pairs (start, destination) in our data set and calculated a mean similarity score for the fixed  $w$ .

We proceeded in the same way for each value of  $w$  in the interval of  $[0, 25]$  (see Figure 1). For  $25 < w \leq 100$  we changed the step size to 10. Should a higher similarity be observed here than was calculated for weights in range 0 to 25, the scope of weights that are sampled in intervals of 1 was extended. This served to ascertain that the best possible results did not lie beyond the boundary that was established.

The maximum detour from the shortest path in order to satisfy user preferences that would still be considered acceptable amounted to  $w = 100$  meters. However, as can be seen in Figures 1 and 2 the similarity between calculated paths and user preferred routes was maximal for  $w < 25$  for each of the factors considered.

### 4 Results

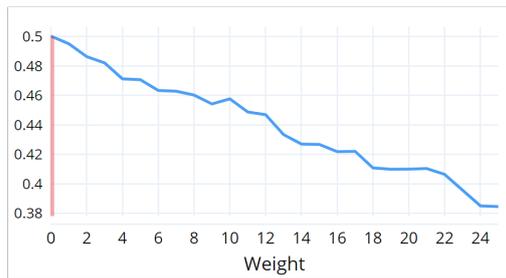
The work was conducted to find answers to the following questions, the answers to which are laid out in corresponding subchapters:

1. Does the consideration of the individual criteria, which were found to be influential to human wayfinding in previous work, lead to a higher similarity to paths that align with human preferences compared to the shortest path? And if so, to what extent?
2. Assuming independence, how does a combination of all criteria with their respective optimal weights impact path similarity?
3. How do the criteria found to be influential compare to one another regarding their impact on human path selection?

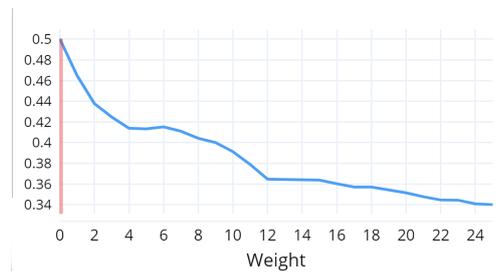
#### 4.1 Individual Evaluation

As described in the preceding chapter, for each factor, the average Similarity score was calculated in dependence on its weighting in order to determine which weight would lead to the best possible average approximation of the entire set of 221 user paths. Starting off the weighting with an increase of 1 in the range of 0 to 25, or if needed higher, provided the weight which leads to the optimum result. The unaltered shortest path displayed a similarity of 50% to the user paths.

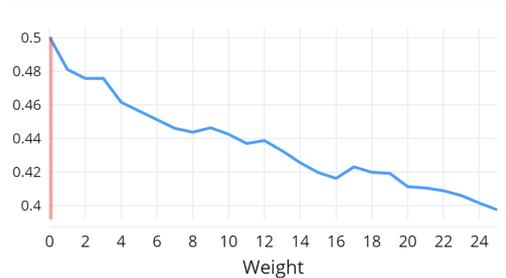
As shown in the following figure, not all factors led to an increase in similarity regardless of the added weight. This proved to be the case for the following factors: decision points, branching factors, doorways, staircases, minimum angle to the destination, and linearity at decision points. For the 5 Results 28 sake of a uniform presentation, all plots encompass a sampling of weights in the range of 0 to 25.



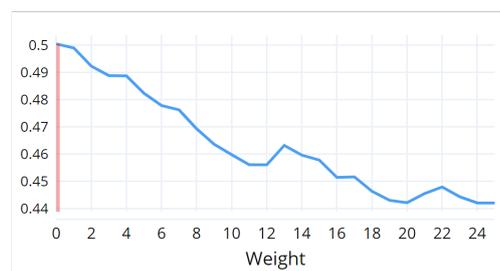
(a) Similarity Scores of Angles.



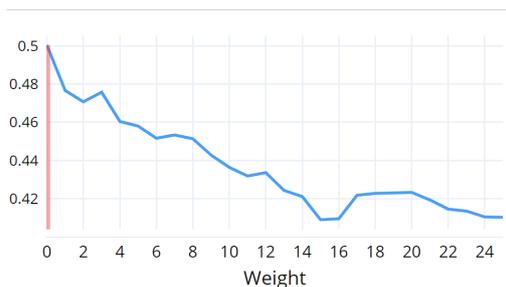
(b) Similarity Scores of Branching Factor.



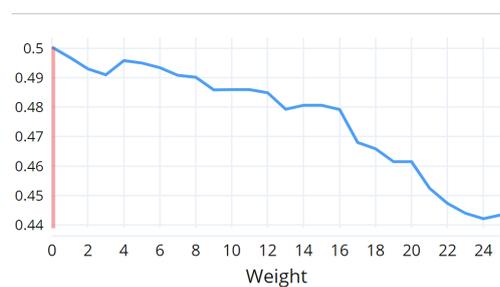
(c) Similarity Scores of Decision Points.



(d) Similarity Scores of Doorways.



(e) Similarity Scores of Linearity.



(f) Similarity Scores of Staircases.

■ **Figure 1** Similarity Scores of Noninfluential Criteria.

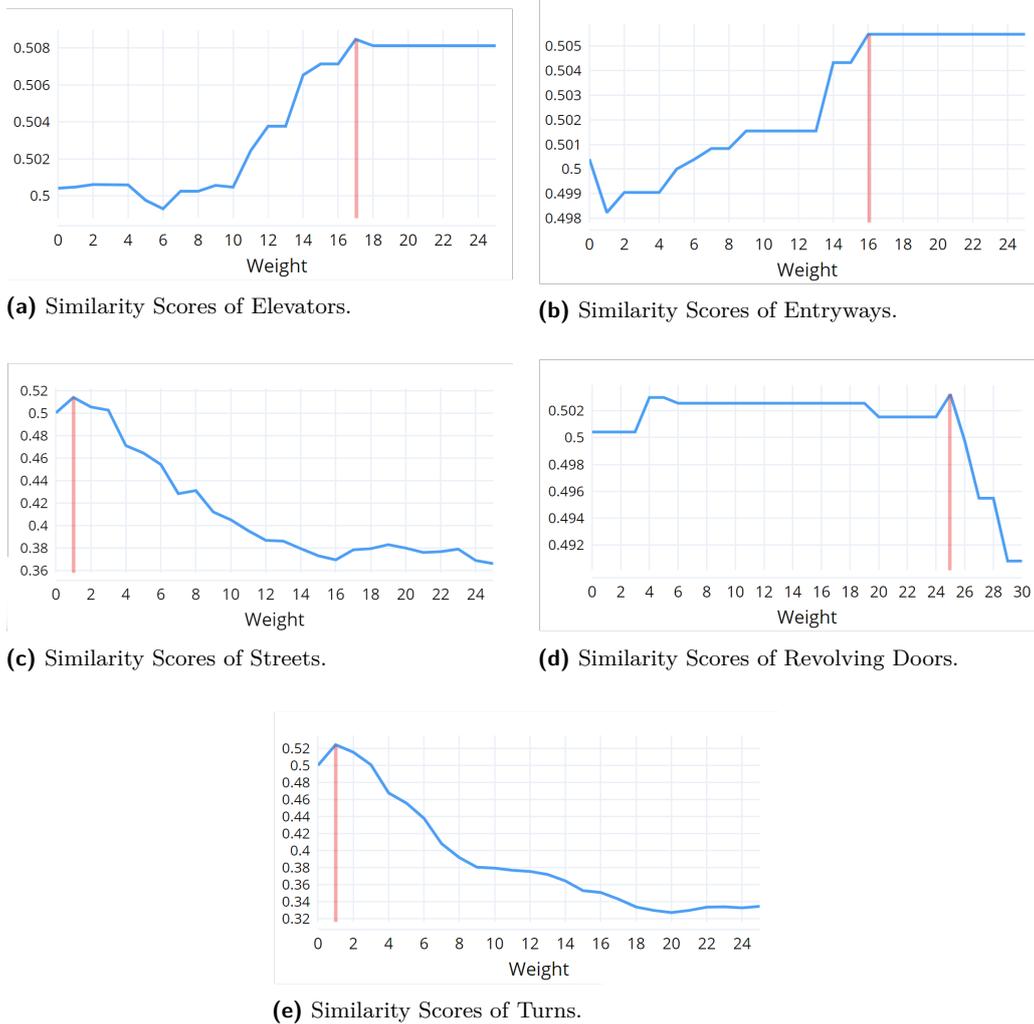
An improvement of the similarity means was therefore reached for all the other inspected criteria, which were turns, streets, entryways, revolving doors, and elevators. The plots depicted in the plots figure 2 visualize the relation between the weights and their improvement, a perpendicular red line marking the weight belonging to the corresponding maximum similarity score.

## 4.2 Combination

Assuming independence, all criteria that have been found to bring about a positive change in similarity score in the preceding section were then combined into one algorithm, each with their established best weight.

Due to the complementary nature of the factors turns and streets, which both penalize slightly altered variations of a deviation from a straight path and are thus evidently not independent from each other, taking into account both of them would lead to duplicate

## 11:10 User Preferences and the Shortest Path



■ **Figure 2** Similarity Scores of Influential Criteria.

weights being added for some edges; only turns were therefore considered in this step since a bigger increase in similarity was observed. With an absolute similarity score of 52.30%, this combination of criteria also resulted in a better approximation of user paths.

### 4.3 Comparison

Table 1 lists the results for each factor and their combination in descending order of resemblance to the user paths measure by similarity score. Additionally, the percentage of affected paths is displayed.

The greatest increase in similarity measured 2.43% and was achieved by turns, i.e. a reduction in the number of turns spanning 90 degrees or less, followed by the combination of criteria with 2.26%. The avoidance of revolving doors produced the smallest improvement of 0.28%.

■ **Table 1** Sorted Table of Criteria with Scores and Percentage of Impacted Paths.

Factors	Weight	Similarity Score	Difference to SP	Impacted Paths
Turns	1	52.47%	2.43%	19.91%
Combination	varied	52.30%	2.26%	32.58%
Streets	1	51.48%	1.38%	19.46%
Elevator	17	50.85%	0.81%	8.60%
Entryway	16+	50.55%	0.51%	11.76%
Revolving Door	25	50.32%	0.28%	2.26%

## 5 Discussion

In consideration of the individual factors, one must retain that not in each case it was a higher number of occurrences that increased or decreased the assessed similarity to user paths. Following the elaboration of the criteria's specific implementation in 4.3, it was their avoidance instead that was implied.

Many of the considered influential factors, previously found to be impactful, did not lead to the computation of paths that were more similar to those people prefer. This was the case for paths that integrated the avoidance of decision points, their reduced branching factors, the avoidance of entrances to rooms, as well as staircases, and a preference for choosing paths that either minimized the global angular distance to the destination's location or aligned with the most linear path selection option. Consequently, all other factors – the avoidance of elevators, entryways, revolving doors, turns, and streets – produced an increase in similarity.

A comparison shows that not only did the criteria's impact on the similarity to user paths vary, but so did the number of affected paths: the differences between the “best” factor (turns) and “worst” one (revolving doors) are almost ten-fold regarding both similarity (2.43% vs. 0.28%) as well as impacted paths (19.91% vs. 2.26%). This certainly may be explained by the frequency with which each factor or entity occurs, for to produce significant changes it has to be present in the paths initially suggested by the shortest path algorithm.

A preference for the most impactful factor can be reinforced by the finding that with an increasing number of turns included in a route, the likelihood of making a mistake while navigating, and thus the time needed for task completion, increases [37]. The erroneous perception that routes containing more turns are longer than routes with fewer turns despite being equal in length (Dalton, 2003), might further explain our results.

In the cases of revolving doors, elevators, and entryways a boundless increase of their weights lead to an improvement (Figures X), signifying that the banishing of all occurring entities would be in the interest of users. Unsurprisingly, since a study conducted on part of the same data set already discovered a significant decline in the number of entryways in preferred routes [36], it was found that adding weights for all occurring entryways would improve the similarity.

The absence of entryways in the favored paths can be understood by considering the inconvenience this would imply, for taking a shortcut through lecture halls would generally only serve as an option during breaks, or passing through shops or offices would be equally unreasonable. A case against revolving doors can be made by considering that these types of doors are generally not barrier-free and users of wheelchairs would naturally avoid them.

The avoidance of choosing elevators as a means of transitioning to other floors implies that people favored the alternative option of taking stairs, which also concurs with previous findings [24, 36].

## 11:12 User Preferences and the Shortest Path

The results found are much the same for turns and streets, which can be explained with the apparent similarities of the factors. Since turns only penalized right-angled deviations, and reducing the number of streets happened by penalizing all deviations from a straight path, reducing the number of streets can be viewed as an extension of turns since above 90 degrees (excluding 180 degrees) were also included.

Despite apparent similarities, a preference for linearity was not found and neither was a preference for minimizing the angular distance. These results could be attributed to decision points, which by themselves, and consequently their branching factor as well, were found ineffective. In addition, although such preferences have formerly been identified [8, 48], the environments in which the studies took place differed greatly from navigating a university campus.

To find the optimal integration of multiple criteria into one algorithm, their specific weights would need to be identified anew. As was expected given the incorrectly assumed independence of the variables, the results fell short of reducing the number of turns alone. For taking dependencies between the factors into account, as a next step, we will train a predictor for  $wlength(x, y)$  – i.e. a neural network that allows non-linear function approximation.

Given all these limitations and observations from our analyses, we can still claim that our algorithm finds a good compromise between shortest paths and user preferred routes: The objective of minimizing path distance, which was evident in every route calculation performed, achieved conformity with the user paths of 50% on average without further augmentation, and can therefore attest to align with human preferences. The overall results are encouraging for the incorporation of the criteria, which were found to lead to an increase of the similarity score, to create an algorithm able to suggest paths that humans like to follow.

## 6 Conclusion

This work aimed to acknowledge users as the focal point of navigational support systems and therefore suggest paths that match with their preferences. To quantify the extent to which this is accomplished, a set of paths, which people take in their daily lives, was considered representative of human preferences, and similarity to these paths was treated as a measure of accomplishment. The assumption that people enjoy being suggested the same routes that they take in well-known environments might be flawed, since aspects regarding the navigation app, such as the ease with which instructions can be displayed and comprehended, are disregarded. Some of the factors were not found to be influential in an indoor environment specifically, but rather in settings whose similarity to the one considered here seems dubious. Narrowing down the findings of previous studies to research applicable to an indoor setting might help identify influential factors more successfully. On the upside, putting these findings into a different context sheds light on their transferability and may offer insights into why they have been found to impact wayfinding decisions. Future research might proceed by taking criteria into consideration that have not been discussed and considered here, including not only environmental factors but more subjective criteria which might be more difficult to assess. Lastly, shedding more light on the interrelations of influential factors might be necessary for the conception of an ideal algorithm that can consider more than one element at a time.

As far as the comparison between indoor and outdoor environments is concerned, our study reproduces the results reported in [22] for indoor environments: user prefer routes with a small number of turns. However, as indoor environments are more complex than streets outside, also other criteria are influential. It is an interesting challenge for future work whether

a transformation of a graph in which nodes actually represent actual distances physical properties of the environment to another model in which nodes represent inevitable choices (such as taking a turn) is feasible if multiple criteria have to be considered simultaneously.

---

## References

- 1 Imad Afyouni, Cyril Ray, and Claramunt Christophe. Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science*, 1(4):85–123, 2012.
- 2 Gary L Allen. Cognitive abilities in the service of wayfinding: A functional approach. *The Professional Geographer*, 51(4):555–561, 1999.
- 3 Iro Armeni and Konstantinos Chorianopoulos. Pedestrian navigation and shortest path: Preference versus distance. In *Intelligent environments (workshops)*, pages 647–652, 2013.
- 4 Michael Brown and James Pinchin. Exploring human factors in indoor navigation. In *The European Navigation Conference*, 2013.
- 5 Beatrix Brunner-Friedrich and Verena Radoczky. Active landmarks in indoor environments. In *International Conference on Advances in Visual Information Systems*, pages 203–215. Springer, 2005.
- 6 David Caduff and Sabine Timpf. On the assessment of landmark salience for human navigation. *Cognitive processing*, 9(4):249–267, 2008.
- 7 Ruth Alison Conroy. *Spatial navigation in immersive virtual environments*. PhD thesis, Citeseer, 2001.
- 8 Ruth Conroy Dalton. The secret is to follow your nose: Route path selection and angularity. *Environment and Behavior*, 35(1):107–131, 2003.
- 9 Daniel Delling, Andrew V Goldberg, Moises Goldszmidt, John Krumm, Kunal Talwar, and Renato F Werneck. Navigation made personal: Inferring driving preferences from gps traces. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, pages 1–9, 2015.
- 10 Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- 11 Matt Duckham and Lars Kulik. “simplest” paths: automated route selection for navigation. In *International Conference on Spatial Information Theory*, pages 169–185. Springer, 2003.
- 12 Ioannis Giannopoulos, Peter Kiefer, Martin Raubal, Kai-Florian Richter, and Tyler Thrash. Wayfinding decision situations: A conceptual model and evaluation. In *International Conference on Geographic Information Science*, pages 221–234. Springer, 2014.
- 13 Sabine Gillner and Hanspeter A Mallot. Navigation and acquisition of spatial knowledge in a virtual maze. *Journal of cognitive neuroscience*, 10(4):445–463, 1998.
- 14 Reginald G Golledge. Path selection and route preference in human navigation: A progress report. In *International conference on spatial information theory*, pages 207–222. Springer, 1995.
- 15 Joy Goodman, Phil Gray, Kartik Khammampad, and Stephen Brewster. Using landmarks to support older people in navigation. In *International Conference on Mobile Human-Computer Interaction*, pages 38–48. Springer, 2004.
- 16 E Grum and AU Frank. Risk of getting lost: Optimize a path to minimize risk. corp, vienna. *Competence Center for Urban and Regional Planning*, 2005.
- 17 Viddit Haria, Yashvi Shah, Vanya Gangwar, Vansh Chandwaney, Tarang Jain, and Yash Dedhia. The working of google maps, and the commercial usage of navigation systems. *International Journal of Innovative Research in Technology*, 6(5), 2019.
- 18 Christoph Hölscher, Tobias Meilinger, Georg Vrachliotis, Martin Brösamle, and Markus Knauff. Up the down staircase: Wayfinding strategies in multi-level buildings. *Journal of Environmental Psychology*, 26(4):284–299, 2006.
- 19 Christoph Hölscher, Thora Tenbrink, and Jan M Wiener. Would you follow your own route description? cognitive strategies in urban route planning. *Cognition*, 121(2):228–247, 2011.

- 20 Alycia M Hund, Martin Schmettow, and Matthijs L Noordzij. The impact of culture and recipient perspective on direction giving in the service of wayfinding. *Journal of Environmental Psychology*, 32(4):327–336, 2012.
- 21 Oliver Jan, Alan J Horowitz, and Zhong-Ren Peng. Using global positioning system data to understand variations in path choice. *Transportation Research Record*, 1725(1):37–44, 2000.
- 22 Bin Jiang and Xintao Liu. Computing the fewest-turn map directions based on the connectivity of natural roads. *International Journal of Geographical Information Science*, 25(7):1069–1082, 2011. doi:10.1080/13658816.2010.510799.
- 23 Daniel Kahneman. Maps of bounded rationality: Psychology for behavioral economics. *American economic review*, 93(5):1449–1475, 2003.
- 24 Panayotis Kikiras, Vassileios Tsetsos, and Stathes Hadjiefthymiades. Ontology-based user modeling for pedestrian navigation systems. In *ECAI 2006 Workshop on Ubiquitous User Modeling (UbiqUM)*, Riva del Garda, Italy, pages 1–6, 2006.
- 25 Alexander Klippel, Heike Tappe, Lars Kulik, and Paul U Lee. Wayfinding choremes—a language for modeling conceptual route knowledge. *Journal of Visual Languages & Computing*, 16(4):311–329, 2005.
- 26 Shohei Koide and Masami Kato. 3-d human navigation system considering various transition preferences. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 859–864. IEEE, 2005.
- 27 Benjamin Kuipers. Modeling spatial knowledge. *Cognitive science*, 2(2):129–153, 1978.
- 28 Carol A Lawton. Gender and regional differences in spatial referents used in direction giving. *Sex Roles*, 44(5):321–337, 2001.
- 29 Kevin Lynch. *The image of the city* (vol. 11), 1960.
- 30 Bharadwaj RK Mantha, Carol C Menassa, Vineet R Kamat, and Clive R D’Souza. Evaluation of preference-and constraint-sensitive path planning for assisted navigation in indoor building environments. *Journal of Computing in Civil Engineering*, 34(1):04019050, 2020.
- 31 David M Mark. Automated route selection for navigation. *IEEE Aerospace and Electronic Systems Magazine*, 1(9):2–5, 1986.
- 32 Richar E Mayer. Visual aids to knowledge construction: Building mental representations from pictures and words. In *Advances in psychology*, volume 108, pages 125–138. Elsevier, 1994.
- 33 Tobias Meilinger, Markus Knauff, and Heinrich H Bühlhoff. Working memory in wayfinding—a dual task experiment in a virtual city. *Cognitive Science*, 32(4):755–770, 2008.
- 34 Daniel R Montello. Spatial orientation and the angularity of urban routes: A field study. *Environment and Behavior*, 23(1):47–69, 1991.
- 35 Daniel R Montello and Martin Raubal. Functions and applications of spatial cognition. In *Handbook of Spatial Cognition*, page 249–264. American Psychological Association, 2013.
- 36 Manuel Müller, Christina Ohm, Florin Schwappach, and Bernd Ludwig. The path of least resistance. *KI-Künstliche Intelligenz*, 31(2):125–134, 2017.
- 37 Michael J O’Neill. Effects of familiarity and plan complexity on wayfinding in simulated buildings. *Journal of Environmental Psychology*, 12(4):319–327, 1992.
- 38 Arto Puikkonen, Ari-Heikki Sarjanoja, Merja Haveri, Jussi Huhtala, and Jonna Häkkinen. Towards designing better maps for indoor navigation: experiences from a case study. In *Proceedings of the 8th international conference on mobile and ubiquitous multimedia*, pages 1–4, 2009.
- 39 Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 116–125, 2014.
- 40 Kai-Florian Richter and Alexander Klippel. You-are-here maps: Wayfinding support as location based service. *GI-Technologien für Verkehr und Logistik. Beiträge zu den Münsteraner GI-Tagen*, 20:21, 2002.
- 41 Edward K Sadalla and Daniel R Montello. Remembering changes in direction. *Environment and Behavior*, 21(3):346–363, 1989.

- 42 Wilson Sakpere, Michael Adeyeye-Oshin, and Nhlanhla BW Mlitwa. A state-of-the-art survey of indoor positioning and navigation systems and technologies. *South African Computer Journal*, 29(3):145–197, 2017.
- 43 Masashi Soeda, Noriko Kushiya, and Ryuzo Ohno. Wayfinding in cases with vertical motion. *Proceedings of MERA*, 97:559–564, 1997.
- 44 Robert J Sternberg. *Metaphors of mind: Conceptions of the nature of intelligence*. Cambridge University Press, 1990.
- 45 Mathieu Taillade, Bernard N’Kaoua, and H el ene Sauz eon. Age-related differences and cognitive correlates of self-reported and direct navigation performance: the effect of real and virtual test conditions manipulation. *Frontiers in Psychology*, 6:2034, 2016.
- 46 YingLi Tian, Xiaodong Yang, Chucai Yi, and Aries Ardit. Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. *Machine vision and applications*, 24(3):521–535, 2013.
- 47 turn. *Cambridge Dictionary*. Cambridge University Press, 2021. URL: <https://dictionary.cambridge.org/dictionary/english/turn>.
- 48 Alasdair Turner. The role of angularity in route choice. In *International Conference on Spatial Information Theory*, pages 489–504. Springer, 2009.
- 49 Manuel Ullmann. Datengetriebene optimierung pr afferenzadaptiver fu swegrouuten durch geb udekomplexe, September 2020. URL: <https://epub.uni-regensburg.de/43697/>.
- 50 Ann Vanclooster, Nina Vanhaeren, Pepijn Viaene, Kristien Ooms, Laure De Cock, Veerle Fack, Nico Van de Weghe, and Philippe De Maeyer. Turn calculations for the indoor application of the fewest turns path algorithm. *International Journal of Geographical Information Science*, 33(11):2284–2304, 2019.
- 51 Ann Vanclooster, Pepijn Viaene, Nico Van de Weghe, Veerle Fack, and Philippe De Maeyer. Analyzing the applicability of the least risk path algorithm in indoor space. In *ISPRS Acquisition and Modelling of Indoor and Enclosed Environments 2013 (Indoor 3D)*, volume 2, pages 19–26. International Society for Photogrammetry and Remote Sensing (ISPRS), 2013.
- 52 Liping Yang and Michael Worboys. Similarities and differences between outdoor and indoor space from the perspective of navigation. *Poster presented at COSIT*, 2011.
- 53 John Zacharias. Pedestrian behavior pedestrian behavior and perception in urban walking environments. *Journal of planning literature*, 16(1):3–18, 2001.
- 54 Shanjiang Zhu and David Levinson. Do people use the shortest path? an empirical test of wardrop’s first principle. *PloS one*, 10(8):e0134322, 2015.



# Map Matching for Semi-Restricted Trajectories

**Timon Behr** ✉

University of Konstanz, Germany

**Thomas C. van Dijk** ✉ 

University of Bochum, Germany

**Axel Forsch** ✉ 

University of Bonn, Germany

**Jan-Henrik Haurert** ✉ 

University of Bonn, Germany

**Sabine Storandt** ✉

University of Konstanz, Germany

---

## Abstract

We consider the problem of matching trajectories to a road map, giving particular consideration to trajectories that do not exclusively follow the underlying network. Such trajectories arise, for example, when a person walks through the inner part of a city, crossing market squares or parking lots. We call such trajectories semi-restricted. Sensible map matching of semi-restricted trajectories requires the ability to differentiate between restricted and unrestricted movement. We develop in this paper an approach that efficiently and reliably computes concise representations of such trajectories that maintain their semantic characteristics. Our approach utilizes OpenStreetMap data to not only extract the network but also areas that allow for free movement (as e.g. parks) as well as obstacles (as e.g. buildings). We discuss in detail how to incorporate this information in the map matching process, and demonstrate the applicability of our method in an experimental evaluation on real pedestrian and bicycle trajectories.

**2012 ACM Subject Classification** Information systems → Geographic information systems

**Keywords and phrases** map matching, OpenStreetMap, GPS, trajectory, road network

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.12

**Supplementary Material** *Software*: <https://github.com/tcvdijk/tesa>  
archived at `swh:1:dir:265a254805a64ab73036cdebd99b59590052a471`

**Funding** *Timon Behr*: DFG Grant Sto 1114/4-2 in Priority Programme 1894 “Volunteered Geographic Information”.

*Thomas C. van Dijk*: DFG Grant Di 2161/2-2 in Priority Programme 1894 “Volunteered Geographic Information”.

*Axel Forsch*: DFG Grant Ha 5451/7-1 in Priority Programme 1894 “Volunteered Geographic Information”.

## 1 Introduction

Map matching is the process of pinpointing a trajectory (given e.g. as a sequence of GPS measurements) to a path in an underlying network. The goal is to find the path that explains the observed measurements best. Map matching is often the first step of trajectory data processing as there are several benefits when dealing with paths in a known network instead of raw location measurement data:

- Both location measurements and geometric representations of roads are usually imprecise. Hence, constraining the trajectory to a path in a network is necessary to integrate the information given with a trajectory (e.g. recorded speed, vehicle type) with information given with the road data (e.g. speed limit, road type). This is important to enable applications as movement analysis or real-time navigation [17, 28].



© Timon Behr, Thomas C. van Dijk, Axel Forsch, Jan-Henrik Haurert, and Sabine Storandt; licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 12; pp. 12:1–12:16

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 12:2 Map Matching for Semi-Restricted Trajectories

- Storing raw data is memory-intensive (especially with high sampling densities). Paths in a given network on the other hand can be stored very compactly and many indexing methods have been developed that allow huge path sets to be queried efficiently [10, 15, 25].
- Matching a trajectory to the road network enables data mining techniques that link attributes of the road network to attributes of the trajectories. This is used to deduce routing preferences from given trajectories [27, 19, 9, 4]. To analyze and compare multiple trajectories, the trajectories need to be matched to a common network beforehand.

However, if the assumption that a given trajectory was derived from restricted movement in a certain network is incorrect, map matching might heavily distort the trajectory and possibly erase important semantic characteristics. For example, there could be two trajectories of pedestrians who met in the middle of a market square, arriving from different directions. After map matching, not only the aspect that the two trajectories got very close at one point would be lost but the visit of the market square would be removed completely (if there are no paths across it in the given network). Not applying map matching at all might result in misleading results as well, though, as the parts of the movement that actually happened in a restricted fashion might not be discovered and – as outlined above – having to store and query huge sets of raw trajectories is undesirable.

Hence, the goal of this paper is to design an approach that allows for sensible map matching of trajectories that possibly contain on- and off-road sections, which we call *semi-restricted trajectories*. We will show that our approach computes paths that are significantly more compact than the raw trajectory data but at the same time still faithfully represent the original movement.

### 1.1 Related work

Due to its practical relevance, there is a huge body of work on map matching. As we cannot cover all the respective papers here in detail, we will focus on related work most relevant to our envisioned application scenario.

**Map matching for restricted trajectories.** According to a recent survey on map matching [6], existing algorithms can be classified into four categories based on the underlying matching principle: similarity models, state-transition models, candidate-evolving models and scoring models. We will now mainly discuss state-transition models as our approach will fit in this category. Early approaches for map matching relied on purely geometric similarities between the road network and the recorded trajectories [31]. One problem of these approaches is their sensitivity to measurement noise and low sampling rates. In order to overcome this problem, more sophisticated approaches try to model sensible transitions between consecutive states. In this context, e.g. Hidden Markov Models (HMM) [11, 14, 18] and reinforcement learning techniques [20, 33] were applied successfully. Furthermore, the incorporation of guiding assumptions, e.g. that movements most likely follow shortest paths, can help to compute more meaningful matches and also to decrease the running time [7, 13].

**Off-road and free space map matching.** Map matching might also lead to artifacts in case the movement did indeed happen in an underlying network but the network data is incomplete. Hence, several approaches have been developed that still produce high-quality matches in this scenario by allowing the path to contain off-road sections [1, 11, 23]. However, these approaches rely on the assumption that the unrestricted movement sequences are rather short which does not have to be the case when dealing with semi-restricted trajectories. Map



■ **Figure 1** Example of a map with an embedded road network as well as free spaces and obstacles. The blue measurement-based trajectory is best explained by movement along the orange dashed path, which follows roads and free spaces where possible but contains no obstacle intersections.

matching for outdoor pedestrian trajectories with a high degree of freedom often requires additional data to yield good results as e.g. derived from a smartphone accelerometer or compass [22, 24]. In other lines of work, the goal is to model the possible movement network as precise as possible (including e.g. cross walks) to be able to adapt conventional map matching techniques for restricted movement [2]. Most map matching approaches for pedestrians focus on indoor movement, though [26, 32, 34]. But these approaches do not consider the possibility to switch between restricted and unrestricted movement and also do not scale well enough to deal with large networks and rich map context.

**Representation methods for unrestricted trajectories.** Existing methods for storing and indexing raw trajectories are often based on a partitioning of the ambient space e.g. into equisized grid cells or by constructing spatial data structures as CSE-Trees [29], ST-R-trees or TB-trees [21]. But these approaches do not feature any kind of compression and are slow in case large trajectory sets are to be reported in a query. In contrast, map-matched trajectories can be managed much more efficiently in terms of memory and retrieval time using methods as SPNET [15], PRESS [25] or PATHFINDER [10]. To gain these advantages also when dealing with unrestricted trajectories, methods for computing an underlying graph for a given set of trajectories were investigated [5]. This task is closely related to map generation from trajectory sets [12, 16]. However, these methods either discard outlier trajectories completely or produce rather large graphs, which is both undesirable in our application. Furthermore, the computed graphs have to be updated whenever new trajectories arrive. We will discuss methods to produce graphs that cover free space areas in such a way that all trajectories that may traverse said free space are represented sufficiently well. This problem has also been investigated in the context of indoor navigation. One approach is to represent the traversable space of floor plans by their skeleton, incorporating points of interest such as entrances and exits [8].

## 1.2 Contribution

We present a pipeline that can accurately map match semi-restricted trajectories, thereby overcoming several limitations of previous work. We consider the following points to be our main contribution:

- We propose an extended network model capable of handling semi-restricted trajectories. In addition to the general graph representation of linear streets, we add tessellated free-space polygons that represent areas of unrestricted movement (such as marketplaces or parking lots). We discuss how a sensible tessellation can be obtained and provide an open source implementation at <https://github.com/tcvdijk/tessa>.
- We significantly extend the map matching approach presented in [11], which was originally developed to enable map matching on incomplete road network data. First, we make the approach more efficient by incorporating ideas from [7]. More importantly, though, we make the algorithm work on our extended network model. Among other things, this requires a carefully designed penalization scheme for off-road sections.
- Furthermore, we describe how to incorporate obstacles in the tessellation as well as the map matching process. Obstacles are polygons that the map matched path is not allowed to intersect (as e.g. buildings for cycle trajectories). Taking obstacles into account requires additional processing steps but allows us to produce more meaningful results.
- We evaluate our approach in an extensive experimental study on real trajectories. We investigate the accuracy of the computed paths, in particular with respect to the recognition of unrestricted movement sections. Figure 1 shows an example outcome of our map matching pipeline.

## 2 Methodology

In this section we develop our map-matching algorithm for *semi-restricted trajectories*, starting with a baseline method for trajectories restricted to a network (Section 2.1) and extending it to deal with outliers, missing segments in the road data, and matches on designated free spaces (Section 2.2).

### 2.1 A baseline algorithm for trajectories restricted to a network

Our algorithm is based on a state-transition model where each point of a given trajectory is represented by a set of matching candidates – the possible system states, i.e., positions of a moving subject. This is similar to HMM-based map-matching algorithms, which aim to find a sequence of positions maximizing a product of probabilities, each of which corresponds to a system state or a state transition. In contrast, we aim to minimize a sum of energy terms. By introducing weights for the energy terms that can be controlled with interactive sliders, the user of the algorithm has an intuitive tool for finding a good parameter setting.

**Input requirements.** The algorithm works on a trajectory  $T = \langle p_1, \dots, p_k \rangle$  of  $k$  points in  $\mathbb{R}^2$ , which we will call the GPS-points. Additionally, we have a directed, edge-weighted graph  $G = (V, E)$  that models the transport network. Every directed edge  $uv \in E$  corresponds to a directed straight-line segment representing a road segment with an allowed direction of travel. For a road segment that can be traversed in both directions,  $E$  contains the directed edges  $uv$  and  $vu$ . For an edge  $e$ , let  $w(e)$  be its *weight*. In the basic setting, we will use the Euclidean length as the weight, but note that other values are possible.

**System states.** For every GPS-point  $p_i$ , we compute a set of matching *candidates* by considering a disk  $D_i$  of prescribed radius  $r$  around  $p_i$ : we select all road segments intersected by  $D_i$  and pick the point nearest to  $p_i$  on each. If such a point is not a node of  $G$ , we inject it as a new node  $v$ . (This means that we split each directed edge  $uv$  containing  $v$

into two directed edges  $uv$  and  $vw$ ; we distribute the weight of  $uv$  to the edges  $uv$  and  $vw$  proportionally to their lengths.) As a consequence, the set of matching candidates for  $p_i$  is a set of *nodes*  $V_i \subseteq V$ . If  $V_i$  is empty, we discard the GPS-point  $p_i$  and do not consider it for the matching.

**State transitions.** Possible transitions between candidates for two consecutive GPS-points are not modelled explicitly. Instead, they are implicitly modeled with the graph  $G$  by assuming that the transition between any two matching candidates  $u \in V_i$  and  $v \in V_{i+1}$  occurs via a minimum-weight  $u$ - $v$ -path in  $G$ . Accordingly, the matched output path  $P$  is defined by selecting, for  $i = 1, \dots, k$ , one candidate  $v_i$  from the set  $V_i$  and connecting every pair of consecutive nodes in the sequence  $\langle v_1, \dots, v_k \rangle$  via a minimum-weight path.

**Energy model.** To ensure that the output path  $P$  matches well to the trajectory, we set up and minimize an energy function that aggregates a state-based and a transition-based energy:

- The *state-based energy* is  $\sum_{i=1}^k \|p_i - v_i\|^2$ , meaning that the energy increases quadratically with the Euclidean distance between a GPS-point  $p_i$  and the matching candidate  $v_i$  selected for it.
- The *transition-based energy* is  $\sum_{i=1}^{k-1} w(P_{i,i+1})$ , where  $P_{a,b}$  is a minimum-weight  $a$ - $b$ -path in  $G$  and  $w(P)$  is the total weight of a path  $P$  (i.e., the sum of the weights of the edges of  $P$ ). For now, we use the geometric length of an edge as its weight.

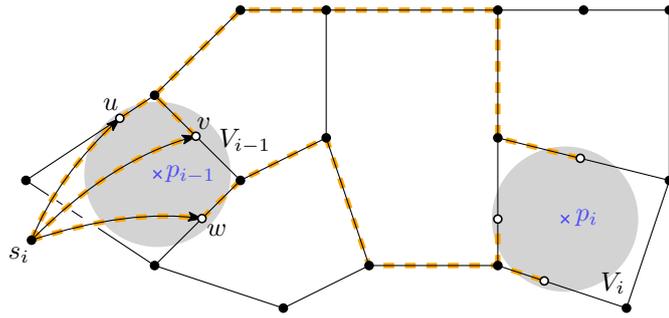
The two energies are aggregated using a weighted sum, parametrized with a parameter  $\alpha_c$ . This yields our overall objective function quantifying the fit between a trajectory  $\langle p_1, \dots, p_k \rangle$  and an output path defined with the selected sequence  $\langle v_1, \dots, v_k \rangle$  of nodes:

$$\text{Minimize } \mathcal{E}(\langle p_1, \dots, p_k \rangle, \langle v_1, \dots, v_k \rangle) = \alpha_c \cdot \sum_{i=1}^k \|p_i - v_i\|^2 + \sum_{i=1}^{k-1} w(P_{i,i+1}) \quad (1)$$

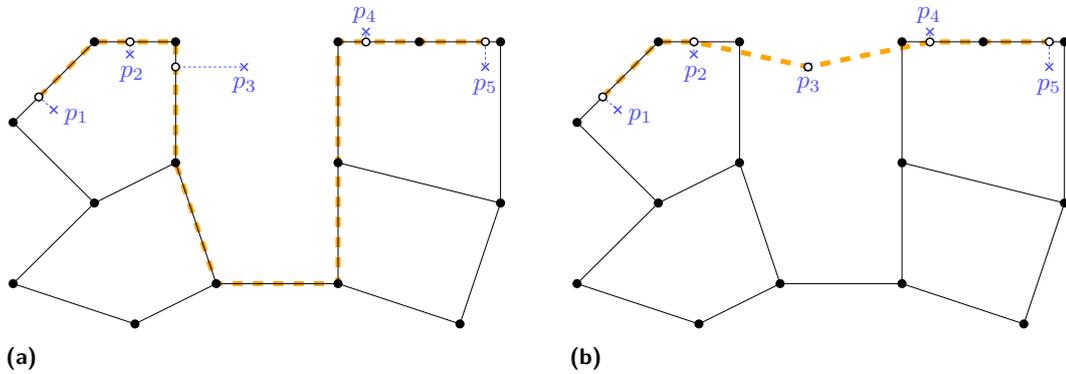
**Algorithm.** An optimal solution is computed using  $k$  runs of Dijkstra's algorithm on a graph that results from augmenting  $G$  with a few auxiliary nodes and arcs. More precisely, we use an incremental algorithm that proceeds in  $k$  iterations. In the  $i$ -th iteration, it computes for the sub-trajectory  $\langle p_1, \dots, p_i \rangle$  of  $T$  and each matching candidate  $v \in V_i$  the objective value  $\mathcal{E}_i^v$  of a solution  $\langle v_1, \dots, v_i \rangle$  that minimizes  $\mathcal{E}(\langle p_1, \dots, p_i \rangle, \langle v_1, \dots, v_i \rangle)$  under the restriction that  $v_i = v$ . This computation is done as follows.

- For  $i = 1$  and any node  $v \in V_1$ ,  $\mathcal{E}_1^v$  is simply the state-based energy for  $v$ , i.e.,  $\mathcal{E}_1^v = \alpha_c \cdot \|p_1 - v\|^2$ .
- For  $i > 1$ , we introduce a dummy node  $s_i$  and a directed edge  $s_i u$  for each  $u \in V_{i-1}$  whose weight we set as  $\mathcal{E}_{i-1}^u$ ; see Fig. 2. With this, for any node  $v \in V_i$ ,  $\mathcal{E}_i^v$  corresponds to the weight of a minimum-weight  $s_i$ - $v$ -path in the augmented graph, plus the state-based energy for  $v$ . We are thus interested in finding for  $s_i$  and every node in  $V_i$  a minimum-weight path. All these paths can be found with one single-source shortest path query with source  $s_i$  and, thus, with a single execution of Dijkstra's algorithm.

We observe that our objective function  $\mathcal{E}(\langle p_1, \dots, p_i \rangle, \langle v_1, \dots, v_i \rangle)$  for the whole trajectory has the minimum value  $\min_{v \in V_k} \{\mathcal{E}_k^v\}$  and is among the values we have computed. It is not difficult to maintain information during the run of the algorithm to enable the reconstruction of a solution attaining that value.



■ **Figure 2** The incremental step of our algorithm. From the dummy node  $s_i$ , shortest paths to all nodes in  $V_i$  are computed with one single-source shortest path query (see the dashed lines). The weight of each such path plus the state-based energy of the corresponding terminal is used as the weight of a dummy edge in the next iteration.



■ **Figure 3** (a) Forcing the input trajectory (blue points) to the road network can cause a long detour in the output path (dashed). (b) Additional unmatched candidates and connecting segments prevent the detour.

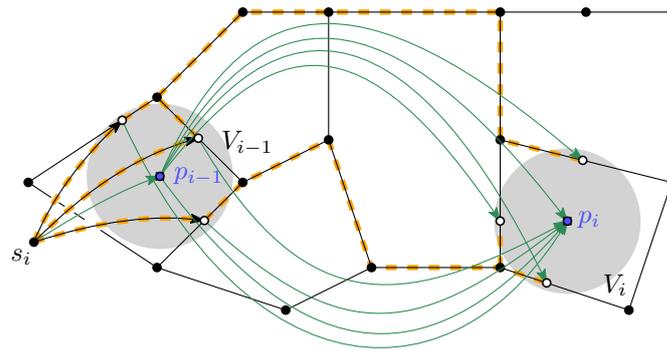
## 2.2 Extensions for semi-restricted trajectories

The algorithm outlined in the previous section forces the output path to the road network, which can lead to unfavorable solutions. Consider the example in Figure 3, where road segments are missing in the road data. By forcing the output path to the road network, a long and unrealistic detour is generated. More generally, we deal with the following issues:

- The trajectory contains outliers.
- The road data is incomplete, i.e., road segments are not modeled in the data.
- The trajectory traverses open spaces that are modeled as areas.

To deal with the first two issues, we allow GPS-points to be left unmatched and accordingly introduce *unmatched candidates*. The third issue is handled by augmenting the road network with additional edges, which we call off-road segments. Accordingly, a matching candidate on an off-road segment is called *off-road candidate*. We present these concepts in the following in detail and then describe how we extend the energy model.

**Unmatched candidates.** For each GPS-point  $p_i$  we extend the set of candidates with a candidate at the observed location. Analogously to the baseline algorithm, we add this candidate into the graph by inserting it as a node  $u_i$ . For every node  $v_{i-1}$  in the candidate set of GPS-point  $p_{i-1}$  and every node  $v_{i+1}$  in the candidate set of GPS-point  $p_{i+1}$  we add



■ **Figure 4** Extended graph to support unmatched segments. In addition to the candidates in the road network, every GPS-point is treated as a candidate itself. Additional arcs (green) are introduced connecting unmatched candidates (blue) to all candidates of the previous and next GPS-point.

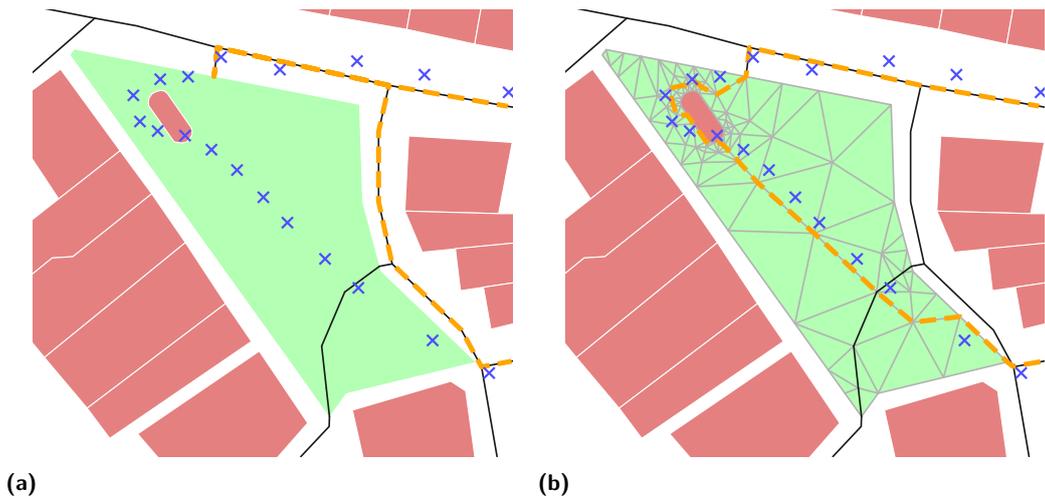
directed edges  $v_{i-1}u_i$  and  $u_iv_{i+1}$ , respectively. Figure 4 shows the extended graph for two consecutive GPS-points with the additional edges shown in green. This approach guarantees that there is always a fallback solution, which can be chosen if no path in the road network is similar to the trajectory. We will associate the additional edges with high energy (see below) to ensure that they will be chosen only if necessary. In the following we refer to the baseline method with the extension to unmatched candidates as Baseline+.

**Off-road candidates.** Open spaces such as public squares, parks or parking lots are represented as polygonal areas in most data sources. The algorithm as described so far does not deal with this (see Figure 5a). We extend the road network by triangulating all open spaces and adding tessellation edges as arcs into the graph (Figure 5b). In order to accurately represent polygons of varying degrees of detail and to provide an appropriate number of off-road candidates, we use CGAL's meshing algorithm [3] with an upperbound on the length of the resulting edges and a lowerbound on the angles inside triangles. As seen in the figure, this algorithm can introduce additional nodes to achieve these constraints. Note that original road segments can cross open spaces: this should be respected by the tessellation since we will give preference to on-road candidates over off-road candidates.

**Extensions of energy model.** With the addition of the extensions to the baseline algorithm we now have three different sets of edges in the graph: the original edges of the road network  $E_r$ , the edges incident to unmatched candidate nodes  $E_u$  and the off-road edges on open spaces  $E_t$ . We prefer on-road matches over off-road matches while unmatched candidates are used as a fallback solution and thus should only be selected if no suitable path over on- and off-road edges can be found. To model this we adapt the energy function by changing the edge weighting  $w$  of the graph. We introduce two weighting terms  $\alpha_t$  and  $\alpha_u$  that scale the weight  $w(e)$  of each edge  $e$  in  $E_t$  or  $E_u$ , respectively. The edge weighting function thus is defined as:

$$w(e) = \begin{cases} \ell(e), & e \in E_r \\ \alpha_u \cdot \ell(e), & e \in E_u \\ \alpha_t \cdot \ell(e), & e \in E_t \end{cases} \quad (2)$$

To favor matches in the original road network and keep unmatched candidates as a fallback solution,  $1 < \alpha_t < \alpha_u$  should hold. Together with the weighting factor  $\alpha_c$  for the state-based energy our final energy function thus comprises three different weighting factors.



■ **Figure 5** (a) Roads in the network data. Movement on the open space (green) cannot be matched appropriately (dashed). (b) Extended road network. Open spaces are tessellated in order to add appropriate off-road candidates. Note that the green polygon has a hole, which remains untessellated since it is not actually open space. Also note that additional vertices are added to prevent skinny triangles.

### 3 Experiments

In this section, we evaluate the presented algorithm on real-world data. For this we conduct a user study in the area of Constance, Germany. In Section 3.1, we describe the used data sources and our experimental setup. In Section 3.2, we present a detailed analysis of the quality of the produced map matching results.

#### 3.1 Experimental setup

In our experiments we used data extracted from OpenStreetMap<sup>1</sup> (OSM) to model the road network. As input trajectories we use trajectories of pedestrians and cyclists recorded within the scope of a user study. In the following we will explain this setup in detail.

**Modelling the road network.** Our experimental region is composed of the area around Lake Constance, Germany. For this region we extract data from OSM to build the model of the road network we use for matching. Elements tagged as highway are used to generate the road network as described in Section 2.1. To make the road network feasible for cyclists and pedestrians we removed all roads that are not traversable for these modes of transport. Altogether we extracted a road graph with 931,698 nodes and 2,013,590 directed edges.

The open spaces used for tessellation are identified by extracting polygons with special tags. We handpicked a list of tags representing spaces with unrestricted movement and tags representing *obstacles* for movement. As the polygons we extracted this way overlapped in some areas we had to sanitize the input: in the case that an obstacle overlapped an open space we cropped the open space to not be covered by the obstacle. In the case that two open spaces overlapped we split the open spaces such that no overlap exists in the final data. This way we extracted 6827 polygons representing open spaces.

<sup>1</sup> © OpenStreetMap contributors

For the tessellation of the open spaces we decided on an upper bound of 25 meters for the length of the resulting edges and kept the lower bound for the inside angles of the triangles at the CGAL default value of about  $20.6^\circ$ . We decided on these parameters by visual inspection of tessellation results. Together with the additional edges of the tessellation our final graph consists of 1,148,213 nodes and 3,345,426 directed edges.

**User study.** To evaluate the performance of the algorithm we conducted a user study with five participants. The participants were asked to record their trips while cycling or walking with their mobile phones or similar GPS-devices. After each trip, the participant annotated all sections of the trip where he or she left roads. This information serves as ground-truth for our evaluation to determine whether these segments got identified correctly. In total, we gathered 58 trajectories during the study. The length of these trajectories varies from 300 meters to 41.3 kilometers, and they contain 66 annotated off-road segments.

### 3.2 Map matching results

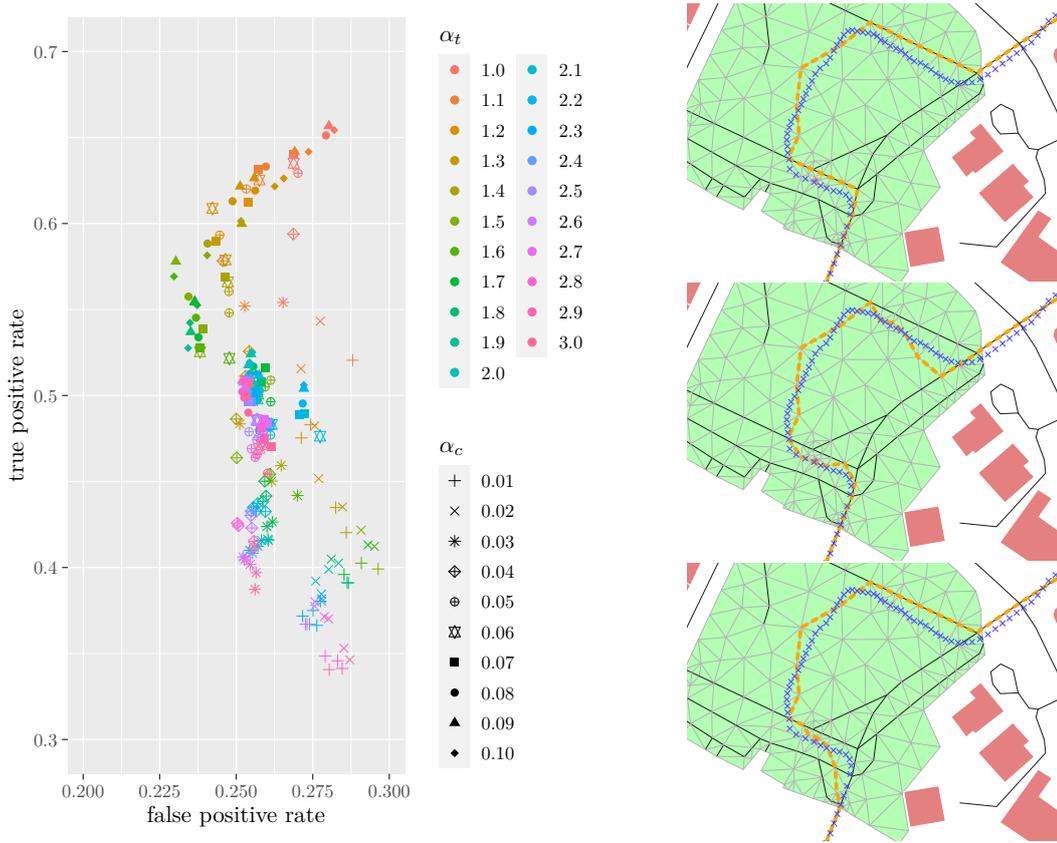
We evaluate our proposed map matching algorithm for semi-restricted trajectories in two steps: In the first step, we investigate the sensitivity of our algorithm to parameter choices in the energy model discussed in Section 2.1 and 2.2, and deduce a sensible parameter setting thereupon. In the second step, we analyze our matching results more thoroughly and compare them to the Baseline+ algorithm without tessellation.

**Parameter tuning.** There are three crucial parameters in our extended energy model that govern the map matching process. First, we have the edge cost coefficients  $\alpha_t$  and  $\alpha_u$ , which for values larger than one penalize the usage of tessellation edges or edges incident to unmatched points, respectively, in comparison to the usage of edges in the road network. Furthermore, the parameter  $\alpha_c$  allows us to fine-tune the impact of the Euclidean distance from a trajectory point to its matched point on the map in the objective function.

As we consider the inclusion of edges incident to unmatched points as a last resort, we set  $\alpha_u$  to 10 in all our experiments. This value is sufficiently large to avoid unmatched points whenever there are road network or tessellation edges in the proximity but it is also small enough to not induce unreasonably long detours if the trajectory indeed crosses an area that has no roads and is also not a free space according to our data set. For  $\alpha_t$  and  $\alpha_c$  the best setting is less obvious, though. For too large  $\alpha_t$  and too small  $\alpha_c$ , the algorithm would match trajectories only to paths in the road network. But a too large value  $\alpha_c$  would render the algorithm too inflexible to find reasonable paths, especially in the presence of outliers.

We hence conducted the following experiment to see how sensitive our algorithm is to the choice of  $\alpha_t$  and  $\alpha_c$ , and to figure out which configuration to use in subsequent studies: We varied  $\alpha_t$  from 1.0 to 3.0 in increments of 0.1 and  $\alpha_c$  from 0.01 to 0.1 in increments of 0.01. For the resulting 310 combinations, we assessed how well the algorithm identifies restricted and unrestricted movement in our labeled trajectory benchmark set. More precisely, we compared the true positive rate of correctly recognized off-road segments (the higher the better) and the false positive rate of actual road parts treated as off-road segments (the lower the better). The results are summarized in Figure 6, left. The effect of certain parameter combinations for the matching of an off-road part is illustrated in Figure 6, right.

A significant range of value combinations is clearly dominated by others, in particular the ones with  $\alpha_t > 2$  or  $\alpha_c < 0.05$ . For  $\alpha_t$  close to 1.0, unsurprisingly, we have the highest true positive rate; but also the highest false positive rate for  $\alpha_t$  in  $[1.0, 2.0]$ . For  $\alpha_c < 0.5$  the true positive rate is rather small. For  $\alpha_c$  chosen larger than that, we see that the precise

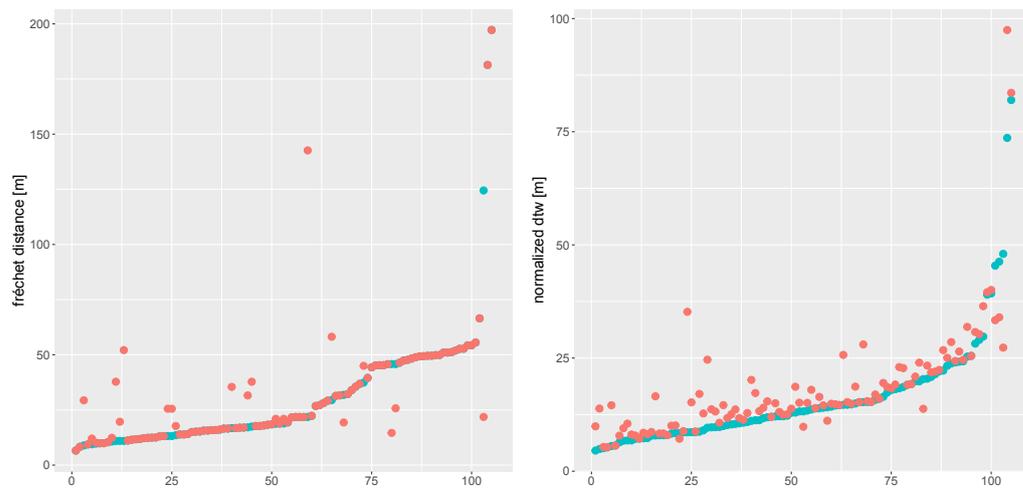


■ **Figure 6** Left: Classification results for off-road section identification for different parameter configurations. Right: Effect of the parameter choice illustrated for an example instance. In the upper image ( $\alpha_t = 1.5, \alpha_c = 0.01$ ), the GPS points (blue crosses) are matched to the cheap road network edges whenever possible. In the middle image ( $\alpha_t = 1.5, \alpha_c = 0.10$ ), the increased candidate cost leads to a matching which deviates from the measurements as little as possible. In the lower image ( $\alpha_t = 1.1, \alpha_c = 0.01$ ), the decreased tessellation edge cost leads to more deviation from the road network as in the upper image. Despite those differences, we observe that the middle part is matched to the same edges for all three configurations.

choice is not that important as  $\alpha_t$  then seems to have a stronger influence on the achieved trade-offs. Furthermore, we computed precision and recall for the identification of on-road sections for all parameter combinations with the results that both, precision and recall, tend to be close to 1.0 for all tested  $\alpha_t$  and  $\alpha_c$ . We hence conclude that the algorithm is not overly sensitive to the precise parameter choice. For subsequent experiments we chose  $\alpha_t = 1.1$  and  $\alpha_c = 0.07$ .

**Quality analysis.** In the parameter tuning experiment, we just considered whether off- and on-road sections were identified as such but now we want to further analyze the overall matching quality, and compare our extended approach for semi-restricted map matching to Baseline+ that does not use tessellated free spaces. We will focus on the following two aspects in our comparative evaluation: (i) shape preservation of the input trajectories, (ii) compression and indexing of trajectory sets.

To measure how well the shape of the original trajectory is preserved by our map matching approach, we compute the Fréchet distance and the Dynamic Time Warping (DTW) distance between the input and the output curve.



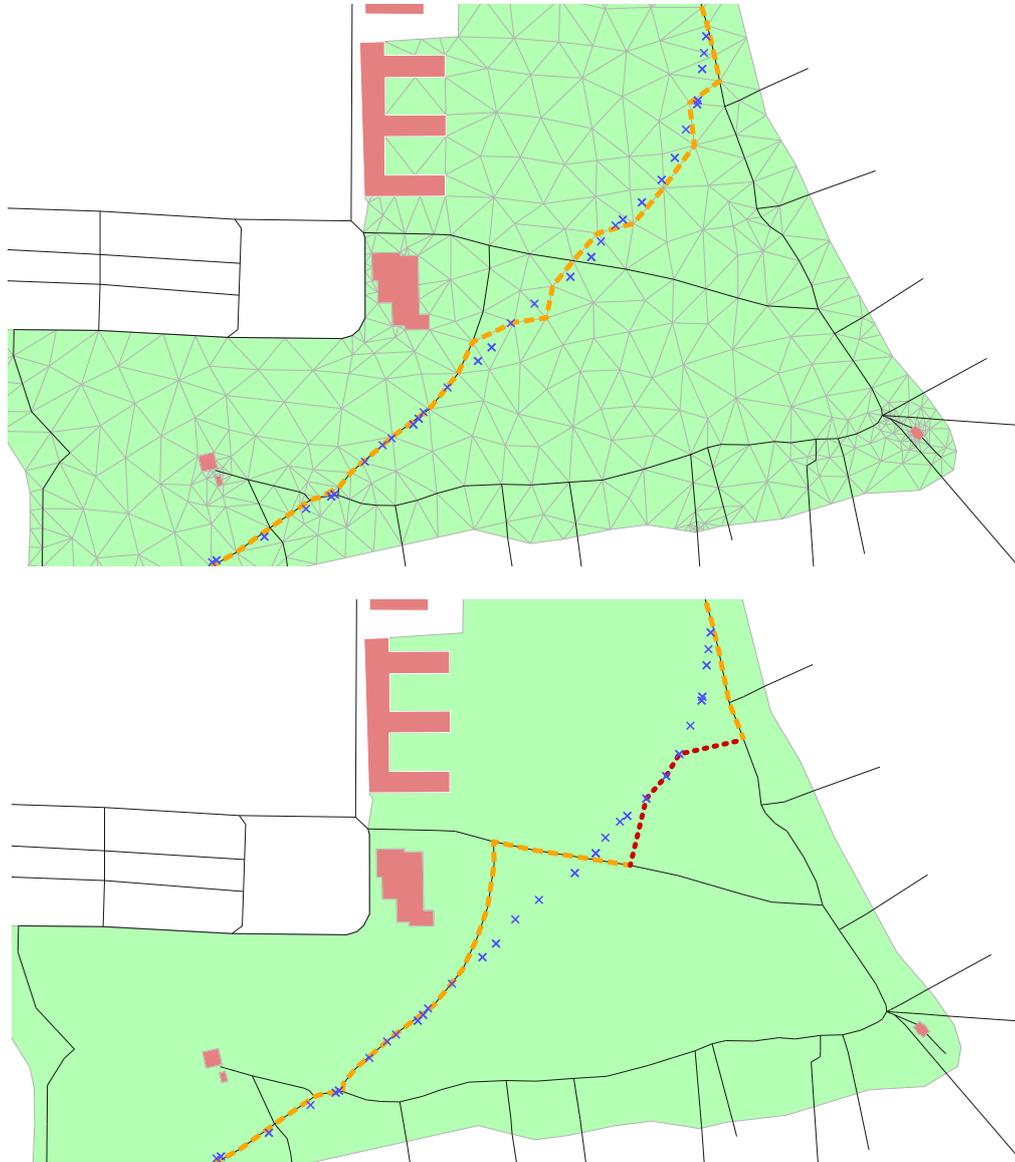
■ **Figure 7** Fréchet and normed DTW distance between input trajectories and the matched paths produced with our approach (blue) and Baseline+ (red). In both cases the trajectories (distributed over the x-axis) are sorted by the fréchet distance (resp. normed DTW) of our approach.

► **Definition 1** (Fréchet Distance). *Given two polylines  $L, L'$  with length  $k$  and  $k'$ , respectively, the Fréchet distance is defined as  $d_F(L, L') := \inf_{\sigma, \theta} \max_{t \in [0, 1]} \|L(\sigma(t)) - L'(\theta(t))\|_2$  where  $\sigma: [0, 1] \rightarrow [1, k]$  and  $\theta: [0, 1] \rightarrow [1, k']$  are continuous and monotonic and  $\sigma(0) = \theta(0) = 1, \sigma(1) = k$  and  $\theta(1) = k'$ .*

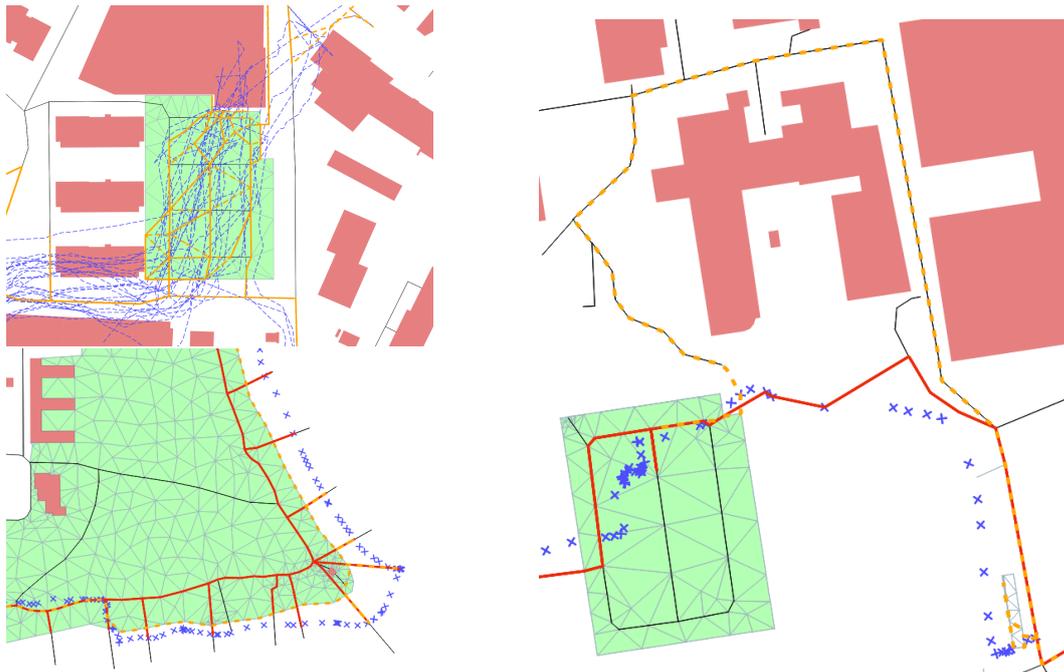
► **Definition 2** (DTW Distance). *Given two polylines  $L, L'$  with length  $k$  and  $k'$ , respectively, the DTW distance is the cost of a cheapest warping path between  $L$  and  $L'$ . A warping path is a sequence  $p = (p_1, \dots, p_W)$  with  $p_w = (l_w, l'_w) \in [1 : k] \times [1 : k']$  for  $w \in [1 : W]$  such that  $p_1 = (1, 1)$  and  $p_W = (k, k')$ , and  $p_{w+1} - p_w \in \{(1, 0), (0, 1), (1, 1)\}$  for  $w \in [1 : W - 1]$ . Thereby, the cost of path  $p$  is defined as  $\sum_{i=1}^W \|p_i\|_2$ . The normed DTW distance is the cost of the path divided by  $2 \cdot \max\{k, k'\}$ .*

The Fréchet distance is a bottleneck measure that sometimes is even used as the main objective for map matching [30]. DTW is frequently used in similarity analysis of time series and other sequenced data, and has the advantage that its value is determined by the whole shapes and not just by a local dissimilarity maximum. We always use the normed DTW distance here, as it allows for better comparability between input trajectories of different length. Figure 7 show the Fréchet and DTW distances for all off-road sections identified in the user study; map matched by our algorithm as well as Baseline+. Although Baseline+ is more likely to keep original trajectory points in the map matched path whenever free spaces are traversed – and hence the shape therein should be perfectly preserved – we observe that our tessellation based approach achieves comparable or even better shape similarity on most inputs. On average, paths computed with our approach had a Fréchet/DTW distance to the original trajectory of 31.81/16.25 while for Baseline+ the respective values are 32.52/18.47. A possible reason for the better shape preservation with our approach is that transitions between restricted and unrestricted movement tend to be smoother, see Figure 8 for an example.

Next, we want to further evaluate whether the tessellation based addition of roughly 1.3 million edges to the road network is worthwhile not only for shape preservation and unrestricted movement identification but also for compression purposes. To this end, we



■ **Figure 8** Original GPS measurements (blue crosses) and matched paths produced by the tessellation based approach (upper image) and Baseline+ (lower image). In the tessellation based matching the shape of the original trajectory is faithfully preserved and the unrestricted movement is correctly identified and matched to tessellation points and edges. Using Baseline+, the first part of the unrestricted movement is wrongfully matched to the path network and subsequently the shape is locally distorted when the switch from restricted to unrestricted movement occurs.



■ **Figure 9** Upper left: Set of trajectories (dashed blue lines) traversing the same free space. The yellow edges indicate their matched paths. Lower left: Trajectory (blue crosses) following the shore line of the lake. The Baseline+ approach matches this trajectory to rather far apart paths as well as several groynes, while the tessellation based approach produces a more sensible match. Right: In the absence of sufficient tessellation edges, a trajectory traversing a large free space might end up being matched to a path with a rather large detour.

count holes in the matched paths. Here, a hole is a continuous subpath that is not represented by edges in the created graph but uses edges incident to unmatched points. The usage of Baseline+ led to 43% more holes than the tessellation based approach. Furthermore, the holes resulting from Baseline+ are typically significantly longer. Holes require additional storage and indexing effort as the contained edges are unlikely to be used by any other trajectories. In contrast, tessellation edges may be traversed by many trajectory matches, see Figure 9 (left) for an example. This not only allows for better compression of trajectory sets but also helps to efficiently retrieve and analyze similar trajectories, and to extract common movement patterns within a trajectory set. Figure 9 further shows an example where Baseline+ computes a nonsensical match for a trajectory that follows the shore line but is then forced to follow existing paths, but also an example where our tessellation based approach fails due to a trajectory traversing an open space which was not chosen for tessellation based on the existing OSM tags. However, for most of the tested instance sensible matches that make use of tessellation edges were identified.

#### 4 Conclusion and Future Work

We introduced the notion of semi-restricted trajectories in this paper and proposed a pipeline for map matching such trajectories to a carefully constructed graph consisting of a road network and tessellated free spaces. We showed that the resulting combined graph is only about 50% larger than the road network but enables a faithful representation of restricted

and unrestricted movement at the same time. As evidenced in our experimental analysis, unrestricted movement is in most cases correctly identified as such by our map matching approach. Nevertheless, there are many interesting directions for future work that could help to improve or extend our approach. For example, different tessellation variants could be tested and compared. At the moment we use an upper bound for the edge length allowed in the free space triangulations and a lower bound for the angular resolution. The two bounds could be varied to achieve different precision and compression trade-offs for the matched trajectories. Other tessellation approaches, e.g. based on polygon skeletons, could further help to model natural movement in free spaces. Furthermore, one could also consider trajectories that enter and exit buildings similarly and try to reliably infer the respective transitions from outdoor to indoor or vice versa. At the moment, we treat all buildings as obstacles, but it is also possible to apply our model to the interior of buildings to allow for indoor matching. Finally, while our approach is sufficiently fast with query times below 0.5 seconds per trajectory, more efficient match computation might be possible by improved candidate node selection, especially within the tessellated free spaces.

---

## References

- 1 Priyanka Aggarwal, David Thomas, Lauro Ojeda, and Johann Borenstein. Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions. *Measurement Science and Technology*, 22(2):025205, 2011.
- 2 Yoonsik Bang, Jiyoung Kim, and Kiyun Yu. An improved map-matching technique based on the Fréchet distance approach for pedestrian navigation services. *Sensors*, 16(10):1768, 2016.
- 3 Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, and Mariette Yvinec. Triangulations in CGAL. In *Proc. 16th Annual Symposium on Computational Geometry (SoCG '00)*, pages 11–18, 2000.
- 4 Anna Brauer, Ville Mäkinen, and Juha Oksanen. Characterizing cycling traffic fluency using big mobile activity tracking data. *Computers, Environment and Urban Systems*, 85:101553, 2021.
- 5 Maike Buchin, Bernhard Kilgus, and Andrea Kölzsch. Group diagrams for representing trajectories. *International Journal of Geographical Information Science*, 34(12):2401–2433, 2020.
- 6 Pingfu Chao, Yehong Xu, Wen Hua, and Xiaofang Zhou. A survey on map-matching algorithms. In Renata Borovica-Gajic, Jianzhong Qi, and Weiqing Wang, editors, *Databases Theory and Applications*, pages 121–133, Cham, 2020. Springer International Publishing.
- 7 Jochen Eisner, Stefan Funke, Andre Herbst, Andreas Spillner, and Sabine Storandt. Algorithms for matching and predicting trajectories. In *Proc. 13th Workshop on Algorithm Engineering and Experiments (ALENEX '11)*, pages 84–95, 2011.
- 8 Birgit Elias. Pedestrian navigation - creating a tailored geodatabase for routing. In *2007 4th Workshop on Positioning, Navigation and Communication*, pages 41–47, 2007.
- 9 Stefan Funke, Sören Laue, and Sabine Storandt. Deducing individual driving preferences for user-aware navigation. In *Proc. 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '16)*, pages 14:1–14:9, 2016.
- 10 Stefan Funke, Tobias Rupp, André Nusser, and Sabine Storandt. PATHFINDER: storage and indexing of massive trajectory sets. In *Proc. 16th International Symposium on Spatial and Temporal Databases (SSTD '19)*, pages 90–99, 2019.
- 11 Jan-Henrik Haunert and Benedikt Budig. An algorithm for map matching given incomplete road data. In *Proc. 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '12)*, pages 510–513, 2012.

- 12 Songtao He, Favyen Bastani, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, and Sam Madden. RoadRunner: improving the precision of road network inference from GPS trajectories. In *Proc. 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '18)*, pages 3–12, 2018.
- 13 George R. Jagadeesh and Thambipillai Srikanthan. Fast computation of clustered many-to-many shortest paths and its application to map matching. *ACM Transactions on Spatial Algorithms and Systems*, 5(3):1–20, 2019.
- 14 Hannes Koller, Peter Widhalm, Melitta Dragaschnig, and Anita Graser. Fast hidden Markov model map-matching for sparse and noisy trajectories. In *Proc. 18th IEEE International Conference on Intelligent Transportation Systems (ITSC '15)*, pages 2557–2561. IEEE, 2015.
- 15 Benjamin Krogh, Christian S. Jensen, and Kristian Torp. Efficient in-memory indexing of network-constrained trajectories. In *Proc. 24th ACM SIGSPATIAL international conference on advances in geographic information systems (ACM SIGSPATIAL GIS '16)*, pages 1–10, 2016.
- 16 Daigang Li, Junhan Li, and Juntao Li. Road network extraction from low-frequency trajectories based on a road structure-aware filter. *ISPRS International Journal of Geo-Information*, 8(9):374, 2019.
- 17 Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '09)*, pages 352–361. ACM, 2009.
- 18 Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '09)*, pages 336–343. ACM, 2009.
- 19 Johannes Oehrlin, Benjamin Niedermann, and Jan-Henrik Haunert. Inferring the parametric weight of a bicriteria routing model from trajectories. In *Proc. 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '17)*, pages 59:1–59:4, 2017.
- 20 Takayuki Osogami and Rudy Raymond. Map matching with inverse reinforcement learning. In *Proc. 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*, pages 2547–2553, 2013.
- 21 Dieter Pfoser, Christian S. Jensen, and Yannis Theodoridis. Novel approaches to the indexing of moving object trajectories. In *Proc. 26th International Conference on Very Large Data Bases (VLDB '00)*, pages 395–406, 2000.
- 22 Ming Ren and Hassan A. Karimi. Movement pattern recognition assisted map matching for pedestrian/wheelchair navigation. *The Journal of Navigation*, 65(4):617–633, 2012.
- 23 Yuya Sasaki, Jiahao Yu, and Yoshiharu Ishikawa. Road segment interpolation for incomplete road data. In *Proc. 2019 IEEE International Conference on Big Data and Smart Computing (BigComp '19)*, pages 1–8, 2019.
- 24 Seung Hyuck Shin, Chan Gook Park, and Sangon Choi. New map-matching algorithm using virtual track for pedestrian dead reckoning. *ETRI Journal*, 32(6):891–900, 2010.
- 25 Renchu Song, Weiwei Sun, Baihua Zheng, and Yu Zheng. PRESS: A novel framework of trajectory compression in road networks. *Proceedings of the VLDB Endowment*, 7(9), 2014.
- 26 Ivan Spassov, Michel Bierlaire, and Bertrand Merminod. Map-matching for pedestrians via bayesian inference. In *Proc. European Navigation Conference*, 2006.
- 27 Jody Sultan, Gev Ben-Haim, and Jan-Henrik Haunert. Extracting spatial patterns in bicycle routes from crowdsourced data. *Transactions in GIS*, 21(6):1321–1340, 2017.
- 28 Shun Taguchi, Satoshi Koide, and Takayoshi Yoshimura. Online map matching with route prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):338–347, 2018.
- 29 Longhao Wang, Yu Zheng, Xing Xie, and Wei-Ying Ma. A flexible spatio-temporal indexing scheme for large-scale GPS track retrieval. In *Proc. 9th IEEE International Conference on Mobile Data Management (MDM 2008)*, pages 1–8, 2008.

## 12:16 Map Matching for Semi-Restricted Trajectories

- 30 Hong Wei, Yin Wang, George Forman, and Yanmin Zhu. Map matching by Fréchet distance and global weight optimization. *Technical Paper, Departement of Computer Science and Engineering*, page 19, 2013.
- 31 Christopher E. White, David Bernstein, and Alain L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.
- 32 Pawel Wilk and Jaroslaw Karciarz. Optimization of map matching algorithms for indoor navigation in shopping malls. In *Proc. 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN '14)*, pages 661–669. IEEE, 2014.
- 33 Ethan Zhang and Neda Masoud. Increasing GPS localization accuracy with reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- 34 Lijia Zhang, Mo Cheng, Zhuoling Xiao, Liang Zhou, and Jun Zhou. Adaptable map matching using PF-net for pedestrian indoor localization. *IEEE Communications Letters*, 24(7):1437–1440, 2020.

# Automated Georeferencing of Antarctic Species

**Jamie Scott** ✉ 

Massey University, Auckland, New Zealand

**Kristin Stock** ✉ 

Massey Geoinformatics Collaboratory, Massey University, Auckland, New Zealand

**Fraser Morgan** ✉ 

Manaaki Whenua Landcare Research, Auckland, New Zealand

**Brandon Whitehead** ✉ 

Manaaki Whenua Landcare Research, Auckland, New Zealand

**David Medyckyj-Scott** ✉ 

Manaaki Whenua Landcare Research, Auckland, New Zealand

---

## Abstract

Many text documents in the biological domain contain references to the toponym of specific phenomena (e.g. species sightings) in natural language form “In <LOCATION> Garwood Valley summer activity was 0.2% for <SPECIES> Umbilicaria aprina and 1.7% for <SPECIES> Caloplaca sp. ...”

While methods have been developed to extract place names from documents, and attention has been given to the interpretation of spatial prepositions, the ability to connect toponym mentions in text with the phenomena to which they refer (in this case species) has been given limited attention, but would be of considerable benefit for the task of mapping specific phenomena mentioned in text documents.

As part of work to create a pipeline to automate georeferencing of species within legacy documents, this paper proposes a method to: (1) recognise species and toponyms within text and (2) match each species mention to the relevant toponym mention. Our methods find significant promise in a bespoke rules- and dictionary-based approach to recognise species within text (F1 scores up to 0.87 including partial matches) but less success, as yet, recognising toponyms using multiple gazetteers combined with an off the shelf natural language processing tool (F1 up to 0.62).

Most importantly, we offer a contribution to the relatively nascent area of matching toponym references to the object they locate (in our case species), including cases in which the toponym and species are in different sentences. We use tree-based models to achieve precision as high as 0.88 or an F1 score up to 0.68 depending on the downsampling rate. Initial results outperform previous research on detecting entity relationships that may cross sentence boundaries within biomedical text, and differ from previous work in specifically addressing species mapping.

**2012 ACM Subject Classification** Computing methodologies → Information extraction; Computing methodologies → Classification and regression trees; Applied computing → Life and medical sciences

**Keywords and phrases** Named Entity Recognition (NER), Taxonomic Name Extraction, Relation Extraction, Georeferencing

**Digital Object Identifier** 10.4230/LIPIcs.GIScience.2021.II.13

**Funding** *Jamie Scott*: Research for this project was funded by Manaaki Whenua Landcare Research.

## 1 Introduction

A significant amount of biodiversity knowledge is locked up in textual descriptions within documents, often in free text form, without coordinates with which to georeference. Many research papers in the biological domain refer to specific species and their location, and the development of methods to extract species-toponym pairs can enable biodiversity mapping, with a range of related societal and economic benefits.



© Jamie Scott, Kristin Stock, Fraser Morgan, Brandon Whitehead, and David Medyckyj-Scott; licensed under Creative Commons License CC-BY 4.0

11th International Conference on Geographic Information Science (GIScience 2021) – Part II.

Editors: Krzysztof Janowicz and Judith A. Verstegen; Article No. 13; pp. 13:1–13:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

However, textual relations can be complex with one-to-many relationships between entities and relationships occurring sometimes across sentence boundaries, and even some distance from each other. To illustrate, Example 1 is a snippet from a journal article [29] containing four relationships between four different species mentions (one just as a broader genus term and the other two in abbreviated G. species form) to the same toponym one, two or three sentences away.

► **Example 1.** “There are also examples of growth forms that make use of the increasing precipitation at <LOCATION> Livingston Island. Most obvious is the very extensive growth of the <SPECIES> *Usnea* lichens whose fruticose form allows them to benefit from both rain, snow and fog (Rundel 1978). Within the mosses, <SPECIES> *A. gainii* and <SPECIES> *H. crispulum* are able to achieve high activities by growing as clumps on the rock surface and storing water from precipitation, and <SPECIES> *A. gainii* is much warmer than other species because of its darker colour.”

A significant body of work has addressed the task of extracting toponym references from documents using named-entity recognition [2, 22], and then disambiguating the mentioned place names to identify the coordinates of the named place, requiring resolution of duplicate place names, place name abbreviations or complex place name sequences [5, 21]. Similarly, efforts have been made to develop automated methods to identify biological species in text [3, 17]. However, little attention has been given to the task of matching specific species mentions to their toponym references, and thus georeferencing (identifying the coordinate location) for specific species mentioned in text documents. Without this step, while we can map toponyms mentioned in a document (as for example in [25] and [1]), we cannot map specific items mentioned in text. In the case of biological text documents, this means that we cannot map species distribution or conduct spatial analysis using data “locked away” in text documents.

The task of georeferencing specific phenomena in text has been addressed in some other domains, including disaster management for event georeferencing [13], and significant attention has been given to extracting spatial relations and the reference and located objects to which they refer [18]. However, the former relies on multiple mentions of the same event, using clustering to converge on its probable toponym, and the latter relies on spatial prepositions to connect located and reference objects. A range of work in the relation extraction field of NLP is also relevant, but until recently the focus has been on relationships that occur within the same sentence [7, 33]. Recent studies that also look for relations that may cross sentence boundaries [28, 8] do not include toponym or species and may have limited transferability, or identify relations that express location in a non-geographic context [19].

There remains a gap in solving the difficult problem of associating species mentions with locations for georeferencing. We address this gap by proposing a method that has three steps:– a) extracting species mentions in text; b) extracting and disambiguating toponym mentions in text and c) predicting which, if any, pairs of those species and toponyms represent an actual <species> ‘present in’ or ‘found at’ <location> relationship. To perform the latter step, we use a machine learning classifier, in which we classify all species-toponym pairs in the document to identify those that are correct matches. We test and evaluate a range of features for our classifier on a corpus of seven documents (44037 tokens) from the Antarctic domain, including a mixture of journal papers, theses and supplementary material, written by authors from around the world.

Preliminary results using tree-based classifiers achieved precision of up to 0.88. This compares favourably to studies within the biomedical field where precision scores have varied from 0.39 to 0.65 [27, 8, 32, 19]. Furthermore, our bespoke rules- and dictionary-based approach for extraction of species achieved an F1 score of 0.87 including partial matches.

The contributions of our work are two-fold. Firstly, we apply machine learning to solve the difficult and unaddressed problem of associating species mentions with locations for georeferencing. Relation extraction in natural language is regarded as a hard problem generally, and little progress has been made on solving it for the fundamental GIScience task of georeferencing the detailed content of documents, with [1, p.1] pointing out that “Research on this problem is still in its infancy”. Generic methods for relation extraction have not been applied to the particular problem of matching species and locations, and cannot handle text where the subject and object of the relation are widely separated, and that involve complex co-references. Furthermore, text that describes location differs from generic language in a number of ways, making the challenge more difficult. Specifically:

1. The ultimate goal of this work is to georeference species mentions, and thus location mentions identified by NER must be tied to specific locations using gazetteers. However, many locations identified by NER are not found in gazetteers, and thus the texts contain many items that look like location references but are actually ‘red herrings’ (not georeferencable), imposing additional demands on standard relation classifiers.
2. The role of toponyms as universal labels for locations presents different language uses, including common but unusual abbreviations (e.g. MDV for McMurdo Dry Valleys), as well as more standard forms (e.g. Mt for Mount).
3. The role of toponyms as a reference frame results in common repetition of place names (or even only once at the beginning of the paper), interspersed with other place names, and the need to disentangle the correct place name references for multiple mentions of different species is especially challenging.

Our second contribution is the method of application of machine-learning to the problem. Our contribution is in the binary classification model using pairwise matches (e.g. in contrast to dependency driven approaches), and the selection of features.

This paper is structured as follows: Section 2 reviews relevant literature, Section 3 outlines our methodology, Section 4 presents results, which are then discussed in Section 5, before conclusions are summarised in section 6.

## 2 Related Work

### 2.1 Recognising Species and Toponyms as Named Entities

The task of detecting location mentions in text is addressed by standard named entity recognition (NER) methods of popular modern natural language processing (NLP) Python packages such as Natural Language Toolkit (NLTK)<sup>1</sup> and spaCy<sup>2</sup>. Recent evaluation of six NER tools on Twitter content indicates precision in the low 90s, with F1 values in the 70s (as recall figures are typically not as high as precision) [16], with slightly lower values in a review on text documents [10]. Following NER, place names must also be resolved to identify the coordinates of the places to which they refer (known as toponym resolution or disambiguation), a task that is challenging due to duplicate place names, unusual abbreviations and spelling variations. Methods for resolving toponyms have included selecting the place with the highest population on the basis that it is more likely to be the one referred to due to size; associated place names mentioned in surrounding text; feature types and language models [4, 15].

---

<sup>1</sup> <http://www.nltk.org>

<sup>2</sup> <https://spacy.io>

## 13:4 Automated Georeferencing of Antarctic Species

In contrast to location, the task of *species* detection is not specifically addressed in standard NER tools. The linguistic structure of taxonomic names does however, allow for the development of automated methods to find these names within natural language text [17]. Linnaean rules for taxonomic names of organisms dictate genus names precede species (and any sub species) name with the former capitalised and all names in either Latin or Greek [17, 23]. Where abbreviated in text, the species name is preceded by the capitalised first letter of the genus and then a period.

Others in the biodiversity domain have tackled the species detection challenge with a mixture of approaches based on matching terms in a dictionary of known entities, [9, 20, 24], isolating taxonomic-looking strings not otherwise in a English lexicon [17, 24], or machine learning approaches like Naïve Bayes [3] or ensemble classifiers that include neural networks [24].

All those methods mentioned above focus on finding scientific names apart from [9] which also finds some common names with their method.

Rule and pattern-based methods can recognise new scientific names in text [17] but can generate false positives for non-scientific text that appears to be in a Linnaean format [24]. Words found in both scientific names and vernacular text cause problems for dictionary and machine learning approaches alike, while machine learning is sensitive to text encoding algorithms [24].

Some dictionary approaches [20] can find new name combinations from existing terms but all are limited by how comprehensive and up-to-date the dictionaries are at time of use, something especially noteworthy in the biodiversity field where thousands of species are discovered or reclassified each year [3, 17, 24].

Recall and precision figures higher than 0.8 and up to 0.97 have been shown to be possible when testing these methods on biodiversity datasets with Quaesitor[24], TaxonFinder[20], and NetiNeti[3] generally performing best in tests [24] conducted.

Despite the collective and respective informativeness of their approaches, none of the options reviewed were suitable for our pipeline for one or more reasons, including the following:

- They were not available as an easy-to-access Python package.
- They returned a single instance of a species mention instead of every mention along with position in the document.
- They separately annotated species and genus for combined terms.
- While available via an online interface or API, they were not appropriate to process entire documents of text, let alone a large collection, due to limits on the amount of text that could be parsed in each use (and while references to DOIs would remove the need to upload entire documents in some cases, a focus of our wider project is processing large batches of legacy documents, some of which do not have DOIs).

### 2.2 Georeferencing Phenomena in Text

The problem of georeferencing of text has been addressed in other domains. Methods for georeferencing documents by identifying their geographic location have been reviewed by [25], but these methods identify a toponym for the entire document, rather than toponyms for specific items mentioned in the document text. Methods have been developed for extracting place mentions in a document addressing a particular topic, assuming that they describe locations relating to the topic. For example, [1] identifies locations of orchards and cancer cases by extracting place names from specific portions of documents (e.g. the methods section), [6] similarly map locations connected to specific historical events across a document,

and [2] apply a similar approach for biological specimens. However, while this approach identifies locations of phenomena, these works do not distinguish between different types of phenomena mentioned in the text, as is our goal.

In the disaster management field, methods to georeference specific events have involved the collection of multiple documents (e.g. news reports) describing a single disaster event, and have clustered locations mentioned in connection with the event to georeference it [13]. However, this approach relies on multiple documents addressing a specific event, rather than individual mentions of a phenomena in a single document, as we address here.

Another significant body of work has focused on extracting place names and more complex place references of biological specimen collections, but this work addresses place references within databases, in which the species that is being located is already known and stored in another database attribute [11, 12, 14], unlike our challenge in which the link between species and location must be identified.

### 2.3 Identifying Relationships

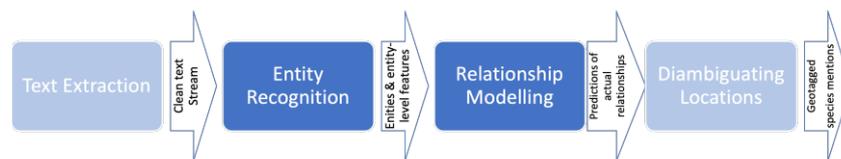
In the Natural Language Processing (NLP) literature, the task of relation extraction is aimed at detecting semantic dependencies between items mentioned in text [26]. A number of relation extraction methods have been developed and increasingly methods have been designed to look for relationships that may extend beyond the sentence boundary [8, 28, 31, 32, 19], since as much as 28-30 per cent of relations in certain corpora are inter-sentential [30, 32], and our corpora contains many examples of complex inter-sentence relations between species and locations, such as that shown in Example 1.

While these methods have been applied before on genetics pathways text using graph Long Short Term Memory (LSTM) [27] and on biochemical text using Bi-affine Relation Attention Networks [32], they have had low success rates, or performed erratically depending on type of entity pair [31]. Other work has addressed only a limited the range of text, concentrating on specific biomedical entity types [27, 28, 32] or using a limited number of adjacent sentences [28]. Recent promising work from [8] looks at a large and varied dataset but one that does not identify species among its named entities. The most comparable work looks at linking bacterial species to habitats (e.g. locations within the human body) within biomedical text [19]. Their method has some commonalities with ours in that they consider syntactic features like the presence of verbs and prepositions, but it does not focus on specific toponyms that can be georeferenced, which is the ultimate goal of this study. Furthermore, while they show significant improvement over baselines, the success metrics achieved are relatively low, demonstrating the challenging nature of the problem.

## 3 Method

### 3.1 Consolidated and Compartmentalised Approaches

The ultimate aim of this project is to lay some of the groundwork for a complete and automated pipeline to extract and process text from legacy documents, identify mentions of species and toponyms within those documents and distinguish which pairs of species-toponym describe an actual geospatial relationship in which a species was found at a particular location. Figure 1 shows the consolidated end-to-end process, but in this paper, we focus on entity recognition and relationship modelling.



■ **Figure 1** Consolidated pipeline process.

## 3.2 Entity Recognition

Whilst different natural language processing packages were considered, SpaCy was chosen over Python’s Natural Language Toolkit (NLTK) for this work, due to its adaptability, especially the option to train new classes for entity recognition via transfer learning which may be useful in future development of this pipeline. To ensure the entity recognition processes were working on clean streams of text, manually pre-cleaned versions of text were copied from PDF formats avoiding images, tables and reference lists to leave only body text, titles and section headings.

### 3.2.1 Species Extraction

Our corpus was entirely within the Antarctic geographic area, and we employed Manaaki Whenua’s<sup>3</sup> list of Antarctic species to assist with species extraction. The list contains over 2100 known Antarctic animals and organisms with full taxonomic names (including kingdom, phylum, class, order, family, genus and species components). For this initial study, we restricted our focus to scientific rather than common names. This restriction, and particularly the Linnean structure of those taxonomic names led us to adopt a rules-based approach to extraction of species mentions.

Working on the basis that a name would appear in text in the form of *Genus species*, *G. species* or possibly just *Genus*, the full list of taxonomic names was reduced and separated into two Python lists of unique genus and species terms of 744 and 1489 terms respectively which were then used as separate look-up lists to match tokens, bigrams or trigrams to full or abbreviated forms of species names in the list, even when the n-gram is in a combination of genus and species terms not previously found together in the original list. The algorithm also identifies mentions of genus only as well as instances where a genus is followed by a term not originally found in the species list but is found elsewhere in the document in the abbreviated G. species form.

### 3.2.2 Toponym Recognition and Disambiguation

In contrast to species, as location is a standard entity in the NER tools of NLP packages including SpaCy’s, our approach for these entities built upon these available tools.

Using SpaCy’s largest English model we applied the NER tool to tokenised documents to identify potential toponyms. These were then checked in a range of gazetteers, with particular emphasis on New Zealand and Antarctic gazetteers (due to the area of interest origin of research) to filter out misidentified toponyms and to pave the way for a genuine toponym resolution process in the future which would link confirmed toponyms to coordinates and thus enable species to be georeferenced.

<sup>3</sup> <https://www.landcareresearch.co.nz>

The gazetteers used were the New Zealand Gazetteer maintained by Land Information New Zealand<sup>4</sup>, the Scientific Committee on Antarctic Research (SCAR) gazetteer<sup>5</sup> and the GeoNames gazetteer<sup>6</sup>. These contained approximately 45,000, 26,000 and 1,500,000 unique place names respectively.

A subset of the New Zealand Gazetteer relating only to toponyms in the New Zealand administered area of Antarctica (approx 5100 toponyms) was created as was a New Zealand-specific subset of the SCAR gazetteer (approx 3,500 toponyms). Similarly, subsets of the GeoNames collection relating to Antarctica (approx 18,000) or New Zealand (approx 45,000) respectively were extracted to create a total of seven gazetteers.

Toponym candidates beginning with “The” had this article removed before searching and toponyms beginning with “Mt.” or “Mt” were standardised as “Mount” to aid with matching. Some one-word place names in the broader GeoNames gazetteer that are also common English words or (e.g. Inner, Upper, Fig) or problematic for this process as a single word (South, North, Mount) were filtered from that gazetteer.

Each toponym was checked against each of the seven gazetteers for exact matches taking special note of toponyms specifically matched in an Antarctic and/or New Zealand gazetteer. All exact matches are put forward to the next stage. Additionally, toponyms not found in either Antarctica or New Zealand gazetteers (including some that were exact matches but only in the vast GeoNames gazetteer, e.g. “Portugal”) are checked for close or partial matches in the more focused New Zealand and Antarctic gazetteers. This is to ensure misspellings, OCR errors, mis-tokenisation or natural variations do not lead to an Antarctic and New Zealand toponym being a) missed, or b) incorrectly linked to a different part of the world (e.g. “Victoria Land” v “Victoria”) as this makes the future step of disambiguation more difficult. We excluded the non-filtered GeoNames list (1.5 million place names) for close matches to focus on matching names to the geographic areas of interest (namely Antarctica and New Zealand) and limit false positives generated during the close and partial matching process.

The close-matching process uses Python’s native `diffib` module<sup>7</sup> to find the best matching entry in each of the six smaller gazetteers if the best match scores over an arbitrarily-set threshold 0.9 using `diffib`’s `get_close_matches` function.

The partial-matching process looks for the biggest sub string within a potential toponym that can be exactly matched in a gazetteer. For example, the five-token candidate “Ross Sea Region of Antarctica” will be broken into two four-token strings (“Ross Sea Region of” and “Sea Region of Antarctica”), three three-token strings (e.g. “Ross Sea Region” and two others) before the algorithm matches “Ross Sea” in a gazetteer once the candidate is broken into two-token sub-strings.

Toponyms confirmed through an exact, close or partial match would then be passed through to the relationship-prediction stage of the pipeline.

### 3.3 Modelling

#### 3.3.1 Conceptualisation of the classification task

The main contribution of the paper is in the method for linking specific species and toponym mentions to each other, given that documents can contain many of each, and related species and toponyms may be spread some distance from each other in the document (see Example 1).

---

<sup>4</sup> <https://gazetteer.linz.govt.nz>

<sup>5</sup> <https://data.aad.gov.au/aadc/gaz/scar/>

<sup>6</sup> [www.geonames.org](http://www.geonames.org)

<sup>7</sup> <https://www.landcareresearch.co.nz>

## 13:8 Automated Georeferencing of Antarctic Species

Furthermore, while every relationship is binary – linking exactly one species instance to one toponym instance in the text – any species or toponyms instance could belong to many relationships. In Example 2: the species instance *Umbilicaria aprina* Nyl. is in two relationships. One with the toponym instance Dry Valleys and another with Botany Bay.

► **Example 2.** Annual activity (% of total time) ranged from 0.2% (*Umbilicaria aprina* Nyl.) in the Dry Valleys (Raggio et al. 2016) through 4.6% for the same species at Botany Bay.

We thus formulate the problem on the basis that each toponym mention in the document is potentially related to every species mention in the same document and vice versa. For example, a document with 100 species mentions and 100 toponym mentions has 10,000 possible actual relationships. We implement this by creating a matrix of all possible species-toponym pairs, and the task is then to identify the actual relationships among the sea of candidates, for which we use a binary classifier.

### 3.3.2 Feature engineering

With the goal of predicting which relationships are genuine among the large number of possibilities, we engineered a range of features (see Table 1) that may indicate whether a species-toponym pair match (i.e. toponym x describes the toponym of species y). Some of these were entity-centric (relating to either the toponym or species) while others helped describe the connection or distance between entities in a potential relationship.

■ **Table 1** Engineered Features.

Feature	Level	Type	Notes
Dependency_Steps	Pairwise	Integer	Length of shortest dependency path
inAbstract300_Toponym	Entity	Boolean	Explanation Below
inAbstract300_Species	Entity	Boolean	Explanation Below
inAbstract500_Toponym	Entity	Boolean	Explanation Below
inAbstract500_Species	Entity	Boolean	Explanation Below
max_TFISF_Toponym	Entity	Float	Explanation Below
max_TFISF_Species	Entity	Float	Explanation Below
Num_Nouns_Between	Pairwise	Integer	Count of nouns between entities
Num_Preps_Between	Pairwise	Integer	Count of prepositions between entities
Num_Tokens_Between	Pairwise	Integer	Count of tokens between entities
Num_Verbs_Between	Pairwise	Integer	Count of verbs between entities
Num_Words_Between	Pairwise	Integer	Count of words between entities
Preposition_Between	Pairwise	Boolean	True if preposition between entities
Same_Sentence	Pairwise	Boolean	True if entities in same sentence
Sent_Start_Toponym	Entity	Boolean	True if entity begins a sentence
Sent_Start_Species	Entity	Boolean	True if entity begins a sentence

#### In Abstract

Four features measured whether or not that entity, regardless of that specific instance’s position in the document, was also mentioned in the abstract of a document as defined by being in either the first 300 or 500 tokens after the first mention of the word “Abstract” in a document. This is an attempt to capture some of the entity’s document-level characteristics with the a priori assumption that if an entity, be it a species or toponym, is mentioned in

the abstract it is a key theme in the document and takes on increased likelihood of being in an actual relationship if mentioned subsequently. This is opposed to for example, that entity mention simply acting as a comparison to a similar species or place for reference, or representing a toponym where downstream aspects of the research, e.g. processing of samples, occurred.

### Term Frequency – Inverse Sentence Frequency (TFISF)

The common measure of term frequency-inverse document frequency (TFIDF) which reflects how important a term is to a document by calculating a term's relative frequency in its own document multiplied by the logged inverse of the proportion of documents it appears in within a corpus. Instead, for our calculations, term frequency – inverse sentence frequency (TFISF) is intended to reflect how important a particular entity is to the sentence containing it.

$$tf(t, s) = f_{(t,s)} \quad (1)$$

$$isf(t, d) = \log\left(\frac{N}{s \in d : t \in s}\right) \quad (2)$$

$$tfisf(t, s, d) = tf(t, s) * isf(t, d) \quad (3)$$

As shown in the equations 1 through 3, term frequency (tf) for a given term (t) is a raw count of how many times that term appears in its sentence (s) (often 1) . Inverse sentence frequency (isf) is the log of the result from dividing the number of sentences in a document (N) by the number of sentences in the document in which the term appears. The two are multiplied together to get TFISF. If an entity contains more than one word, TSISF is calculated for each word in the entity and the highest result selected.

### 3.3.3 Modelling Approaches

Given the modelling task is a binary classification problem with a known ground truth, a range of supervised machine learning classifiers were applied including logistic regression, AdaBoost, a neural network and three tree-based models – random forest, light gradient boosting machine and extra randomised trees. All were imported as python packages from Scikit-Learn or, in the case of the neural network, Keras.

As this was an exploratory assessment of each model's suitability for further testing and since the size of the training data is too small, at least in terms of members of the target class (i.e., actual relationships) to allow for the creation of an adequately-sized validation set in addition to the training and test splits, no parameters were fined-tuned for any models due to the risk of over-fitting to the training data. Only default parameters were used and a broad-brush summary of some of the key parameters follows.

AdaBoost used 50 estimators and a learning rate of 1, while Light GBM's learning rate was set at 0.1. Light GBM, random forest and extra randomised trees all used 100 estimators with no max depth. Gini was used as the criterion for splitting in both random forest and extra randomised trees while light GBM boosting type was set to a traditional gradient boosting tree.

Logistic regression's penalty function was set to L2 and the solver used was limited memory BFGS while the neural network, which could not be run entirely on default settings was constructed from input and one hidden layers of 64 nodes each, both with rectified linear units (ReLU) as the activation functions and an output layer using sigmoidal activation function. The loss function was binary cross entropy.

## 13:10 Automated Georeferencing of Antarctic Species

Some features were transformed for normality for use in logistic regression and the neural network but only untransformed features were used for the tree-based or boosting models as they are not affected by monotonic transformations to data.

### 4 Results

#### 4.1 Data Overview

The data consisted of two tranches of documents with one used to develop the entity recognition algorithms and an annotated second set used to act as the ground truth for measuring the performance of the entity recognition and relationship prediction stages.

The first set contained nine PDF documents ranging from PhD theses to academic journal articles and supplementary material with some documents electronically borne and others scanned from printed versions. As these were not annotated, they remained unsuitable for testing and model building.

The second tranche of seven documents, all electronically borne, was annotated by one of the co-authors, a domain expert, using Tagtog, an online text annotation tool. Firstly, any combination of genus and species (or a genus and species term in isolation) were tagged as species and all toponyms were tagged as such. Next, any relationships between one species and one toponym were tagged if the text indicated the species was present or found in that toponym even if that relationship crossed the sentence boundary. Each tagged relationship was binary, linking exactly one species instance to one toponym instance in the text, but any species or toponyms instance could belong to multiple one-to-one relationships.

■ **Table 2** Training Data Overview.

Doc.	Sentences	Tokens	Species Mentions	Location Mentions	Potential Pairs	True Pairs	True Pair %
1	214	5120	12	96	1152	11	0.955
2	252	7192	13	87	1131	12	1.061
3	335	8779	55	138	7590	48	0.632
4	194	5524	92	35	3220	11	0.342
5	82	2387	3	64	192	4	2.083
6	143	4212	8	31	248	3	1.210
7	323	10823	83	95	7885	40	0.507
Total	1543	44037	266	546	21418	129	0.602

Within this collection (as seen in Table 2), the tokenised documents ranged in length from 2300 tokens to over 10,000 tokens and from just 82 sentences to 335 sentences. Overall, the training data contained 44,037 tokens and 21,418 possible relationships between the 266 species mentions and 546 toponym mentions. This is a smaller than ideal set but indicates potential in the presented methods applied to texts describing object locations, with multiple mentions and relations that may be greatly separated.

#### 4.2 Performance of Entity Recognition

Tables 3 and 4 show the performance of the approaches to recognising species and toponyms within the documents when matched to annotated entities, using five-fold cross-validation. We measured precision, recall and F1 score on two levels. Firstly, if there is an exact match between the entity as it was annotated and as it was extracted by the entity recognition

process (see Table 3); and secondly including partial matches where an extracted or annotated entity is contained entirely within the other (see Table 4). A simple, and easily correctable, example of a partial match for species is the algorithm failing to match the *spp.* token of *Diplosphaera spp.* because SpaCy incorrectly parses the period as a separate token but the process nonetheless recognises and extracts the genus on this example and this may still hold some value for the end user.

■ **Table 3** Species and Toponym Recognition – Exact Matches.

Entity	Precision	Recall	F1
Species	0.9279	0.7256	0.8144
Toponym	0.9760	0.4469	0.6131

■ **Table 4** Species and Toponym Recognition – Exact & Partial Matches.

Entity	Adjusted Precision	Adjusted Recall	Adjusted F1
Species	0.9904	0.7744	0.8692
Toponym	0.9840	0.4505	0.6180

The rules- and dictionary-based approach to species recognition correctly identifies 72.6 per cent of species exactly as they are tagged by the annotator and this recall figure rises to 77.4 per cent when instances of partial matches are included. Precision is higher at 0.9279 (0.9904 including partial matches) with an overall F1 score for the species recognition process of 0.8144 (0.8692).

The corresponding results for the toponym recognition process using SpaCy's inbuilt NER tool and a multi-level gazetteer matching process are generally lower than that for species with the exception of precision for exact matches which is higher at 0.9760. This reflects a high degree of confidence that a toponym that passes through the gazetteer-matching process as either an exact, close or partial match in a gazetteer will also have been tagged as a toponym by the annotator. This figure rises to 0.9840 when partial matches with what has been annotated are included.

However, recall for toponym is 0.4469 (0.4505 including partial matches) and for some documents, this is as low as 0.2286 for exact matches. Some ultra-specific toponyms e.g. subglacial caves like *Harry's Dream* and *22 Blue* are not recognised in gazetteers and abbreviations like *MDVs* for all subsequent mentions of *McMurdo Dry Valleys* in one particular document are filtered out by the gazetteer-matching process as false negatives.

### 4.3 Performance of Species-Toponym Matching Model

We performed classification with three tree-based classifiers, after tests with neural networks, logistic regression and AdaBoost proved less successful.

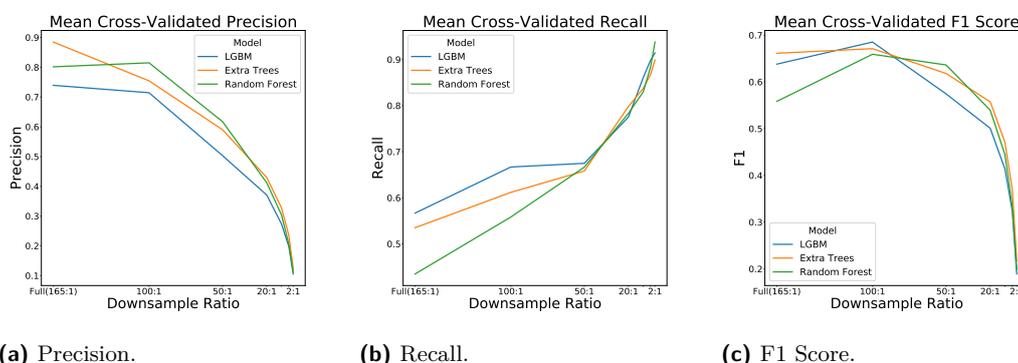
Because of the large ratio of false relationships to actual relationships, we attempt to mitigate the impact by down-sampling the data at different rates to reduce the imbalance in the data. The original ratio of 160 false relationships for every one actual relationship was reduced to 100:1, 50:1, 10:1, 5:1 and 2:1 in different iterations of testing to gauge if this approach could help a machine learning tool make better predictions.

Table 5 shows the average performances of the model and down-sample combinations with three classifiers with five-fold cross validation. Light GBM and extra randomised trees combined with a slight down-sample of 100 to 1 achieved the highest F1 scores, while the extra trees model also did comparatively well on the 'full' training sample within each cross-validation split.

■ **Table 5** Five-fold Mean Cross Validation Scores.

Model	Down-sample Ratio	Precision	Recall	F1
Extra Trees	None (full sample)	<b>0.884561</b>	0.535077	0.661317
Extra Trees	100:1	0.754360	0.611692	0.671119
Extra Trees	50:1	0.590000	0.658154	0.618070
Extra Trees	10:1	0.327795	0.837231	0.470736
LGBM	None (full sample)	0.739273	0.566769	0.638022
LGBM	100:1	0.714369	0.666769	<b>0.685243</b>
LGBM	50:1	0.502949	0.674769	0.574422
LGBM	10:1	0.273593	<b>0.860615</b>	0.413559
Random Forest	None (full sample)	0.801411	0.434769	0.558376
Random Forest	100:1	0.814839	0.558154	0.659163
Random Forest	50:1	0.617086	0.666462	0.636299
Random Forest	10:1	0.304225	0.829538	0.444405

The trends in the effect of down-sampling can further be seen in the graphs in Figure 2 which respectively plot precision, recall and F1 across models and down sampling rate. The more aggressively the majority ‘no relationship’ class was down-sampled to match the minority ‘actual relationship’ class the higher the rate of recall as the models got better at finding all actual relationships. However, this occurred at the expense of precision and generated more false positives. The F1 score tended to increase for a minor down-sampling effort of 100:1 for all three models and fall away after that.



■ **Figure 2** Mean Cross-Validated Metrics by Downsample Rate and Model.

#### 4.4 Feature Importance

For an indication of feature importance in the three types of tree-based models, we extracted the feature importance rankings for the respective models when used on full samples. Importance is determined by the the number of splits using each feature as a percentage of total splits in the the respective collections of decision trees that comprised each model. Table 6 shows some commonalities among which features were used most by the various models.

The variables created to capture whether or not a species was also mentioned in the abstract (at either 300 or 500 tokens from the first mention of ‘Abstract’ in the document) and whether or not a preposition is between the two entities provided little to no value in the

models, but the TSISF variables for both species and toponym were in the top seven features for all three models and were the most important for the LGBM model, while TSISF for toponym was also the most important for the extra randomised trees model. All of the five *number of <type> between* distance variables rounded out the top seven in each model, and the *number of dependency steps* also appeared in the top nine in each of the models.

■ **Table 6** Most Important Features by Model Type.

Feature	Extra Random Trees		Light GBM		Random Forest	
	Rank	Imp.	Rank	Imp.	Rank	Imp.
max_TFISF_Location	1	0.1461	1	0.2687	5	0.1222
Num_Words_Between	2	0.1317	7	0.0553	1	0.1700
Num_Tokens_Between	3	0.1316	4	0.1203	2	0.1530
Num_Preps_Between	4	0.1217	6	0.0787	4	0.1295
Num_Nouns_Between	5	0.1105	5	0.1027	3	0.1401
Num_Verbs_Between	6	0.1095	3	0.1477	6	0.0901
max_TFISF_Species	7	0.0657	2	0.1670	7	0.0721
Dependency_Steps	8	0.0589	9	0.0113	9	0.0266
Sent_Start_Location	9	0.0491	8	0.0133	8	0.05055

## 5 Discussion

### 5.1 Entity Recognition

The precision for both species and toponym recognition is high (>0.9 for both) and minor adjustments to the species method utilising existing SpaCy functionality such as out-of-vocab tags could help it learn to recognise species and genus names that aren't listed in a given dictionary. This would lift the species recall figure which sits at 0.77 including partial matches. Training a NLP engine to identify species entities is another option but not tested so far with this work due to a lack of training data.

With a recall of just 0.45, larger adjustments are required to the toponym recognition process. The task of toponym extraction in these types of documents is challenging because of issues such as “second mentions” of place names (e.g. “McMurdo Dry Valleys”), including pronouns (“it”), shortened forms (“the Valleys”) and acronyms (“MDVs”), and methods to address these through coreference resolution would improve results. Loosening the gazetteer filtering process to allow more toponym candidates to pass through would improve recall, but at the cost of precision.

### 5.2 Relationship Extraction

The performance of the relationship modelling process with limited fine-tuning of tree-based models is promising. The highest precision for predicting actual relationship was 0.88 (for the Extra Trees model with no down-sampling), and the highest recall was 0.86 (for the LGBM model with 10:1 down-sampling). The highest F1 score of 0.68 was achieved for LGBM with 100:1 down-sampling (see Table 5). Further work on larger annotated data sets would allow for validation sets to be created in addition to train/test splits and facilitate tuning of models.

Ours is an approach that has not been applied to species-toponym relationships before and it shows promising results when compared with other work addressing related, but different problems in the biomedical fields, which use graph LTSMs [27], Bi-affine Relation

Attention Networks [32], and transformer-type networks [8]. Furthermore, our work is not limited to finding cross-sentence relations only in adjacent or near-adjacent sentences like [28]. The process has highlighted however, the potential of exploring coreference resolution as recent studies [8, 28] have done.

While some of the engineered features, namely the two TSISF measures showed promise in the modelling process, others seemingly offered little utility. The process of engineering and exploring new features (e.g. word vectors, sentence polarity etc) that help map the relationship of species and toponym should be explored further and may yield improved results as part of future work.

## 6 Conclusion

In this paper, we have described a method to extract mentions of species and place names from text documents, and then to determine which place names describe the toponyms of which species.

We have demonstrated a rules- and dictionary-based approach for the extraction of species in the Antarctic context, and applied existing place name extraction methods, with a set of gazetteers to identify toponyms. Our main contribution is the development of a method that uses tree-based classifiers to match toponyms and species mentions in order to identify the toponym of specific species, for georeferencing and mapping, with a precision of 0.88 (highest F1 of 0.68). In future work, we plan to include additional features in the model, employ a larger corpus for training and tuning; and improve efficiency through filtering some species-toponym combinations (those that are unlikely) before applying the classifier.

This research contributes to the goal of georeferencing text mentions of specific species on two specific fronts. Firstly, little to no research is available on automatically extracting species toponyms from text documents, and the work described in this paper is among the first to provide a method for extracting specific mention toponyms in the biological domain. Secondly, cross-domain methods for georeferencing mentions of different kinds of phenomena (and being able to identify which kinds of phenomena are where) in a document have been limited thus far. While this is early work, the method shows promise, particularly for dealing with relations that may cross sentence boundaries and contain other kinds of complexities such as abbreviations and a many to many relationship between species and toponyms.

---

## References

- 1 Elise Acheson and Ross S. Purves. Extracting and modeling geographic information from scientific articles. *PLOS ONE*, 16(1):e0244918, January 2021. doi:10.1371/journal.pone.0244918.
- 2 Moises Acuna-Chaves and José Araya. Extraction of geographic entities from biological textual sources. In *2017 XLIII Latin American Computer Conference (CLEI)*, pages 1–8, 2017. doi:10.1109/CLEI.2017.8226422.
- 3 Lakshmi Manohar Akella, Catherine N. Norton, and Holly Miller. NetiNeti: discovery of scientific names from text using machine learning methods. *BMC Bioinformatics*, 13(1):211, 2012. doi:10.1186/1471-2105-13-211.
- 4 Ana Bárbara Cardoso, Bruno Martins, and Jacinto Estima. Using recurrent neural networks for toponym resolution in text. In *EPIA Conference on Artificial Intelligence*, pages 769–780. Springer, 2019.
- 5 Arthur D Chapman and John R Wiczorek. *Georeferencing Best Practices*. GBIF Secretariat, Copenhagen, 2020. doi:10.15468/doc-gg7h-s853.

- 6 Rachel Chasin, Daryl Woodward, Jeremy Witmer, and Jugal Kalita. Extracting and displaying temporal and geospatial entities from articles on historical events. *The Computer Journal*, 57(3):403–426, 2014.
- 7 Hong-Woo Chun, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun’Ichi Tsujii. Extraction of Gene-Disease Relations from Medline using Domain Dictionaries and Machine Learning. In *Biocomputing 2006*, pages 4–15, Maui, Hawaii, December 2005. World Scientific. doi:10.1142/9789812701626\_0002.
- 8 Markus Eberts and Adrian Ulges. An End-to-end Model for Entity-level Relation Extraction using Multi-instance Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3650–3660, 2021. URL: <https://www.aclweb.org/anthology/2021.eacl-main.319>.
- 9 Martin Gerner, Goran Nenadic, and Casey M. Bergman. LINNAEUS: A species name identification system for biomedical literature. *BMC Bioinformatics*, 11(1):85, 2010. doi:10.1186/1471-2105-11-85.
- 10 Milan Gritta, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. What’s missing in geographical parsing? *Language Resources and Evaluation*, 52(2):603–623, 2018.
- 11 Qinghua Guo, Yu Liu, and John Wiecek. Georeferencing locality descriptions and computing associated uncertainty using a probabilistic approach. *International Journal of Geographical Information Science*, 22(10):1067–1090, 2008.
- 12 Robert P Guralnick, John Wiecek, Reed Beaman, Robert J Hijmans, the BioGeomancer Working Group, et al. Biogeomancer: automated georeferencing to map the world’s biodiversity data. *PLoS Biol*, 4(11):e381, 2006.
- 13 Felix Hamborg, Corinna Breiter, and Bela Gipp. Giveme5W1H: A Universal System for Extracting Main Events from News Articles. *arXiv:1909.02766 [cs]*, September 2019. URL: <http://arxiv.org/abs/1909.02766>.
- 14 Andrew W Hill, Robert Guralnick, Paul Flemons, Reed Beaman, John Wiecek, Ajay Ranipeta, Vishwas Chavan, and David Remsen. Location, location, location: utilizing pipelines and services to more effectively georeference the world’s biodiversity data. *BMC bioinformatics*, 10(14):1–9, 2009.
- 15 Yiting Ju, Benjamin Adams, Krzysztof Janowicz, Yingjie Hu, Bo Yan, and Grant McKenzie. Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling. In *European Knowledge Acquisition Workshop*, pages 353–367. Springer, 2016.
- 16 Morteza Karimzadeh, Scott Pezanowski, Alan M. MacEachren, and Jan O. Wallgrün. GeoTxt: A scalable geoparsing system for unstructured text geolocation. *Transactions in GIS*, 23(1):118–136, 2019. doi:10.1111/tgis.12510.
- 17 Drew Koning, Indra Neil Sarkar, and Thomas Moritz. TaxonGrab: Extracting Taxonomic Names From Text. *Biodiversity Informatics*, 2, 2005. doi:10.17161/bi.v2i0.17.
- 18 Parisa Kordjamshidi, Martijn Otterlo, and Marie-Francine Moens. Spatial role labeling: Towards extraction of spatial relations from natural language. *TSLP*, 8:4, December 2011. doi:10.1145/2050104.2050105.
- 19 Parisa Kordjamshidi, Dan Roth, and Marie-Francine Moens. Structured learning for spatial information extraction from biomedical text: bacteria biotopes. *BMC bioinformatics*, 16:129, April 2015. doi:10.1186/s12859-015-0542-z.
- 20 Patrick R. Leary, David P. Remsen, Catherine N. Norton, David J. Patterson, and Indra Neil Sarkar. uBioRSS: Tracking taxonomic literature using RSS. *Bioinformatics*, 23(11):1434–1436, 2007. doi:10.1093/bioinformatics/btm109.
- 21 Jochen L. Leidner. Toponym resolution in text: annotation, evaluation and applications of spatial grounding. *ACM SIGIR Forum*, 41(2):124–126, December 2007. doi:10.1145/1328964.1328989.

- 22 Jochen L. Leidner and Michael D. Lieberman. Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Special*, 3(2):5–11, 2011. doi:10.1145/2047296.2047298.
- 23 Carl Linnaeus. *Species Plantarum*. Laurentius Salvius, Stockholm, Sweden, 1753.
- 24 Damon P. Little. Recognition of Latin scientific names using artificial neural networks. *Applications in Plant Sciences*, 8(7):e11378, 2020. doi:10.1002/aps3.11378.
- 25 Fernando Melo and Bruno Martins. Automated Geocoding of Textual Documents: A Survey of Current Approaches. *Transactions in GIS*, 21(1):3–38, 2017. doi:10.1111/tgis.12212.
- 26 Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39, 2021.
- 27 Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-Sentence N-ary Relation Extraction with Graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017. doi:10.1162/tac1\_a\_00049.
- 28 Chris Quirk and Hoifung Poon. Distant Supervision for Relation Extraction beyond the Sentence Boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182, Valencia, Spain, April 2017. Association for Computational Linguistics. URL: <https://www.aclweb.org/anthology/E17-1110>.
- 29 Burkhard Schroeter, T. G. Allan Green, Ana Pintado, Roman Türk, and Leopoldo G. Sancho. Summer activity patterns for mosses and lichens in Maritime Antarctica. *Antarctic Science*, 29(6):517–530, December 2017. doi:10.1017/S095410201700027X.
- 30 Kumutha Swampillai and Mark Stevenson. Inter-sentential Relations in Information Extraction Corpora. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.
- 31 Kumutha Swampillai and Mark Stevenson. Extracting relations within and across sentences. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 25–32, 2011.
- 32 Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi:10.18653/v1/N18-1080.
- 33 Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794, 2015.